Package ‘fistools’

July 17, 2024

Title Tools & data used for wildlife management & invasive species in Flanders

Version 0.2.0

Description This package contains functions & data that are widely used within the wildlife management & invasive species research group (FIS) of the research institute forest and nature (INBO).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports dplyr (>= 1.1.4),
httr (>= 1.4.7),
magrittr (>= 2.0.3),
stringr (>= 1.5.1),
rlang (>= 1.1.3),
progress (>= 1.2.3),
googledrive (>= 2.1.1),
svDialogs (>= 1.1.0),
utils (>= 4.3.2),
uuid (>= 1.2.0),
devtools (>= 2.4.5),
DBI (>= 1.2.3),
sf (>= 1.0.16),
osmdata (>= 0.2.5),
units (>= 0.8.5)

Contents

`apply_grtsdb` *apply grtsdb*

Description

Applies `grtsdb::extract_sample` from inbo/GRTSdb to a custom perimeter. This function installs GRTSdb if it is missing from your machine.

Usage

```
apply_grtsdb(perimeter, cellsize = 100, n = 20, export_path = ".", seed)
```

Arguments

`perimeter`	a simple features (sf) object
`cellsize`	an optional integer. The size of each cell. Either a single value or one value for each dimension. Passed onto extract_sample from GRTSdb. Default is 100.
`n`	an optional integer. the sample size. Passed onto extract_sample from GRTSdb. Default is 20
`export_path`	an optional character string pointing to the path where the GRTSdb.sqlite is created. Default is "."
`seed`	a optional character. Allowing to rerun a previous use.

Details

A function to apply grtsdb to a custom perimeter

GRTSdb is automatically installed when missing from your system.

Author(s)

Sander Devisscher

Examples

```
## Not run:
# Preparation
perimeter <- sf::st_as_sf(boswachterijen$boswachterijen_2024) %>%
  dplyr::filter(Regio == "Taxandria",
                Naam == "vacant 4")

# A new sample
sample <- apply_grtsdb(perimeter,
                       cellsize = 1000,
                       n = 20,
                       export_path = ".")
```

```
leaflet::leaflet() %>%
 leaflet::addTiles() %>%
 leaflet::addCircles(data = sample$samples,
                     color = "red") %>%
 leaflet::addPolylines(data = sample$grid,
                       color = "blue") %>%
 leaflet::addPolylines(data = perimeter,
                       color = "black")
# Reuse a old sample
seed <- sample$seed

sample <- apply_grtsdb(perimeter,
                       cellsize = 1000,
                       n = 20,
                       export_path = ".",
                       seed = seed)

 leaflet::leaflet() %>%
 leaflet::addTiles() %>%
 leaflet::addCircles(data = sample$samples,
                     color = "red") %>%
 leaflet::addPolylines(data = sample$grid,
                       color = "blue") %>%
 leaflet::addPolylines(data = perimeter,
                       color = "black")

## End(Not run)
```

`boswachterijen` *Boswachterijen*

Description

Spatiale en andere informatie (o.a. telefoonummers) van de boswachterijen van ANB.

Usage

```
boswachterijen
```

Format

`boswachterijen`**:**
En sf data frame with 98 rijen and 11 kolommen per jaar:

Regio Beheerregio

Naam Naam van de boswachter

telefoon Telefoon nr van de boswachter ...

Source

https://www.who.int/teams/global-tuberculosis-programme/data

`check` *Check*

Description

Helper script to determine existence in environment panel

Usage

```
check(x)
```

Arguments

`x`	environment object

Details

This doesn't work with functions which will yield a 0 by default.

Value

1 = object exists in environment 0 = object doesn't exist in environment

Author(s)

Sander Devisscher

`cleanup_sqlite` *cleanup sqlite*

Description

A helper script to cleanup after use of apply_gtrsdb.

Usage

```
cleanup_sqlite(db = "grts.sqlite")
```

Arguments

`db`	name of the temporary .sqlite db to be removed

`colcompare` *Columnname comparison*

Description

A simple function to list the difference in column names in 2 datasets.

Usage

```
colcompare(x, y)
```

Arguments

`x`	dataframe 1
`y`	dataframe 2

Value

a list of columns present in x but not in y and a list of columns present in y and not in x.

Author(s)

Sander Devisscher

Examples

```
## Not run:
# create example dataframes
super_sleepers <- data.frame(rating=1:4,
animal=c('koala', 'hedgehog', 'sloth', 'panda'),
country=c('Australia', 'Italy', 'Peru', 'China'),
avg_sleep_hours=c(21, 18, 17, 10))

super_actives <- data.frame(rating=1:4,
animal=c('kangeroo', 'wolf', 'jaguar', 'tiger'),
country=c('Australia', 'Italy', 'Peru', 'China'),
avg_active_hours=c(16, 15, 8, 10))

colcompare(super_sleepers, super_actives)

## End(Not run)
```

`collect_osm_features` *collect OpenStreetMaps features*

Description

Extracts spatial features from the OpenStreetMaps server: features that are extracted are re-classified into broad categories:

- osm_polygons: urban, agriculture, open, forest, water
- osm_lines: roads, waterways
- osm_points: city names

Usage

```
collect_osm_features(
  proj_bbox,
  download_features = "all",
  landuse_elements = "all",
  line_elements = "all"
)
```

Arguments

`proj_bbox`	A bbox. The bounding box for the project/ study area for which to extract osm features.
`download_features`	A character. "all" download all features. "polygons", "lines" and "points" to download only polygon, line or point features respectively. Combinations are also possible (e.g. c("polygons", "points")). Default is "all".
`landuse_elements`	A character. "all" to download all landuse classes. "urban", "agriculture", "open", "forest" and "water" to download only landuse classes of interest. Combinations are also possible (e.g. c("urban", "forest", "water")) Default is "all".
`line_elements`	A character. "all" to download all line elements. "road", "water" to download only roads and rivers, streams etc. respectively. Default is "all"

Details

A function to collect custom osm features for a project

dplyr and osmdata are automatically installed when missing from your system.

Value

a named list of 3 sf data frames: osm_polygons, osm_lines, osm_points. Each sf data frame contains the corresponding geometry types.

Author(s)

Martijn Bollen

Examples

```
## Not run:

# extract the bounding box (WGS84) for the Project
proj_sf <- st_sfc(st_polygon(list(drg_example$spatial$coordinates[1,,])), crs = 4326)
proj_bbox <- st_bbox(proj_sf)
class(proj_bbox)
# extract selected OSM features
osm <- collect_osm_features(proj_bbox)
# extract only polygon OSM features
osm_polygons <- collect_osm_features(proj_bbox, download_features = "polygons")
osm_lines <- collect_osm_features(proj_bbox, download_features = "lines")
osm_points <- collect_osm_features(proj_bbox, download_features = "points")
osm_forest_water <-
 collect_osm_features(proj_bbox, download_features = c("polygons", "points"),
                      landuse_elements = c("forest", "water"))
# extract combination of OSM features, subset line elements
osm_polygons_roads <-
 collect_osm_features(proj_bbox, download_features = c("polygons", "lines"),
                      line_elements = "road")

# calculate the area of each landuse class within the bbox
landuse <- osm_polygons$osm_polygons %>%
 st_make_valid() %>%
 mutate(area = set_units(st_area(.), "km^2")) %>%
 group_by(landuse) %>%
 summarise(area = sum(area))

# plot
## polygons
(p1 <- ggplot(osm_polygons$osm_polygons %>% filter(!is.na(landuse)) %>% arrange(landuse)) +
   geom_sf(aes(fill = landuse), col = NA) +
   scale_fill_manual(values = unique(arrange(osm$osm_polygons, landuse)$osm_fill)) +
   theme_void() + theme(legend.position = "right"))

## lines
(p2 <- ggplot(osm_lines$osm_lines) +
   geom_sf(aes(col = line_element)) +
   scale_color_manual(values = c("grey50", "#0092da")) +
   theme_void() + theme(legend.position = "right"))

## points
(p3 <- ggplot(osm_points$osm_points) + geom_sf_label(aes(label = name)))

## combine features
p1 + geom_sf(data = osm$osm_lines, aes(col = line_element)) +
 geom_sf_label(data = osm$osm_points, aes(label = name)) +
 scale_color_manual(values = c("grey20", "#0092da")) +
 coord_sf(xlim = proj_bbox[c("xmin", "xmax")],
          ylim = proj_bbox[c("ymin", "ymax")])

## End(Not run)
```

`download_dep_media` *Download deployment media*

Description

This function allows the user to download all media related to a Agouti - dataset which matches the given parameters.

Usage

```
download_dep_media(
  dataset,
  depID,
  species = NULL,
  favorite = FALSE,
  outputfolder = NULL
)
```

Arguments

`dataset`	character string, path to the folder where a camptraptor datapackage has been unzipped.
`depID`	character string, ID of the deployment to download media from.
`species`	character string, latin name of the species to download
`favorite`	boolean, do you only want the pretty pictures?
`outputfolder`	character string, path where the function should download the media into

Details

If you are getting an Authorization Error (#403), this probably means your Agouti project has Restrict Images on. This needs to be turned off. If depID = "all" and favorite = TRUE, the function will download all favorited pictures in the whole dataset.

Value

Downloads the specified media files into the outputfolder

Author(s)

Lynn Pallemaerts

Emma Cartuyvels

Sander Devisscher

Soria Delva

Examples

```
## Not run:
drg <- fistools::drg_example

# Situation 1: download whole deployment
download_dep_media(dataset = drg,
                   depID = "96413aa6-5f1f-4dfb-8fab-8f06decc179f")

# Situation 2: download only wanted species
download_dep_media(dataset = drg,
                   depID = "96413aa6-5f1f-4dfb-8fab-8f06decc179f",
                    species = "Dama dama")

# Situation 3: download only favorited species media
download_dep_media(dataset = drg,
                    depID = "96413aa6-5f1f-4dfb-8fab-8f06decc179f",
                    species = "Dama dama",
                    favorite = TRUE)

# Situation 4: download only favorited species media
download_dep_media(dataset = drg,
                    depID = "all",
                    favorite = TRUE)

## End(Not run)
```

download_gdrive_if_missing

Download gdrive if missing

Description

This function downloads the specified file from google drive if the destination file does not exist. If it does exist the user will be prompted to download it again.

Usage

```
download_gdrive_if_missing(gfileID, destfile, update_always = FALSE, email)
```

Arguments

`gfileID`	character google file token
`destfile`	character destination filename with extention
`update_always`	optional boolean to trigger a download everytime the function is run. default is FALSE.
`email`	optional character specifying the users email used to access the googledrive file.

Details

Its best practice to provide the email in encrypted form. This can be easily achieved by adding email as an item in a .renviron file or even beter by using more robust encryption methods.

Value

If the destination file was missing it is now downloaded from the googledrive.

Author(s)

Sander Devisscher

Examples

```
## Not run:
# download newest version of the team charter
download_gdrive_if_missing(gfileID = "1gtqcZojPnbLhEgpul3r9sy2zK3UyyCVG",
                           destfile = "../../Teamcharters/Teamcharter_FIS.pdf",
                           email = Sys.getenv("email"),
                           update_always = TRUE)

## End(Not run)
## Not run:
# download newest DRG Agouti export
download_gdrive_if_missing(gfileID = "1FX8DDyREKMH1M3iW9ijWjVjO_tBH8PXi",
                  destfile = "../fis-projecten/Grofwild/Drongengoed/Input/Agouti/drongengoed_240502.zip",
                           email = Sys.getenv("email"),
                           update_always = TRUE)

## End(Not run)
```

`download_seq_media` *Download sequence media*

Description

This function allows the user to download all media related to a Agouti - sequence which matches the given parameters.

Usage

```
download_seq_media(dataset, seqID, favorite = FALSE, outputfolder = NULL)
```

Arguments

`dataset`	character string, path to the folder where a camptraptor datapackage has been unzipped.
`seqID`	character string, ID of the sequence to download media from
`favorite`	boolean, do you only want the pretty pictures?
`outputfolder`	character string, path where the function should download the media into

Details

If you are getting an Authorization Error (#403), this probably means your Agouti project has Restrict Images on. This needs to be turned off.

Value

Downloads the specified media files into the outputfolder

Author(s)

Lynn Pallemaerts

Emma Cartuyvels

Sander Devisscher

Soria Delva

Examples

```
## Not run:
drg <- fistools::drg_example

# Situation 1: download whole sequence
download_seq_media(dataset = drg,
                    seqID = "f4c049d2-d42f-4cd3-a951-fd485ed0279a")

# Situation 2: download only favorited species media within sequence
download_seq_media(dataset = drg,
                    seqID = "f4c049d2-d42f-4cd3-a951-fd485ed0279a",
                    favorite = TRUE)

## End(Not run)
```

`drg_example` *drg_example*

Description

Subset of Drongengoed Agouti export for testing of functions

Usage

```
drg_example
```

Format

datapackage

deployments Lijst van geselecteerde deployments

observations Observaties van de geselecteerde deployments

media Lijst van media-urls van de geselecteerde deployments

Source

`https://www.agouti.eu`

`label_converter` *label converter*

Description

Script to convert labelnummer, soort en/of labeltype en jaar into afschotlabel

Usage

```
label_converter(
  input,
  id_column,
  labelnummer_column,
  soort_column,
  labeltype_column,
  jaar_column,
  output_style = "eloket"
)
```

Arguments

`input`	a dataframe containing the necessary columns.
`id_column`	a character string pointing to a column used to link result with input.
`labelnummer_column`	a character string pointing to the column containing label numbers.
`soort_column`	a character string pointing to the column containing species.
`labeltype_column`	a character string pointing to the column containing label types.
`jaar_column`	a character string pointing to the column containing years.
`output_style`	a charcter string specifiying the output style. Can be "eloket" or "labo". Default is "eloket".

Details

The input dataframe should a least contain a id_column & labelnummer_column other values can be 'hardcoded'.

Value

a dataframe containing 2 columns id & label

Examples

```
## Not run:

# provide a dataframe with the necessary columns
df <- data.frame(
  id = 1:1000,
  labelnummer = sample(1:1000, 1000, replace = TRUE),
  soort = sample(c("REE", "WILD ZWIJN", "DAMHERT"), 1000, replace = TRUE),
  labeltype = sample(c("REEKITS", "REEGEIT", "REEBOK", NA), 1000, replace = TRUE),
  jaar = sample(2018:2020, 1000, replace = TRUE)
)

labels <- label_converter(df, "id", "labelnummer", "soort", "labeltype", "jaar", "eloket")

# provide a dataframe with labelnummer & labeltype & hardcode soort & jaar
df <- data.frame(
id = 1:1000,
labelnummer = sample(1:1000, 1000, replace = TRUE),
labeltype = sample(c("REEKITS", "REEGEIT", "REEBOK", NA), 1000, replace = TRUE)
```

```
)

labels <- label_converter(df, "id", "labelnummer", "REE", "labeltype", 2020, "eloket")

# provide a dataframe with labelnummer & soort & hardcode labeltype & jaar

df <- data.frame(
id = 1:1000,
labelnummer = sample(1:1000, 1000, replace = TRUE),
soort = sample(c("REE", "WILD ZWIJN", "DAMHERT"), 1000, replace = TRUE))

labels <- label_converter(df, "id", "labelnummer", "soort", "REEKITS", 2020, "eloket")

# provide a dataframe with mixed labelnummers & labeltype & hardcode soort & jaar
df <- labels %>%
  left_join(df %>% select(-labelnummer), by = "id") %>%
  add_row(id = setdiff(1:1000, labels$id)) %>%
 mutate(labelnummer = ifelse(is.na(labelnummer), sample(1:1000, 1000, replace = TRUE), labelnummer)) %>%
 mutate(labeltype = ifelse(is.na(labeltype), sample(c("REEKITS", "REEGEIT", "REEBOK", NA), 1000, replace = TRU

labels <- label_converter(df, "id", "labelnummer", "REE", "labeltype", 2020, "eloket")

# to troubleshoot
df_test <- df[!df$id %in% labels$id,]


## End(Not run)
```

`UUID_List` *UUID list generator*

Description

A helper script to generate a list of UUIDs

Usage

```
UUID_List(temp_input)
```

Arguments

`temp_input`	a data.frame to which UUIDs should be appended

Index