Package ‘sitingclass’

December 14, 2023

Type Package

Title Siting classification of Norwegian weather stations

Version 0.1.0

Author Pierre-Marie Lefeuvre

Maintainer Pierre-Marie Lefeuvre <pierre.marie.lefeuvre@met.no>

Description The package sitingclass evaluates the exposure of weather stations for specific sensors from digital elevation models. The package is currently designed for Met usage only, due to data restrictions.

License GPL3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

R topics documented:

compute_horizon *Horizon height at a station*

Description

Compute horizon height in degrees from a station location in UTM 33

Usage

```
compute_horizon(
  centre = NULL,
  dem = NULL,
  level = 2,
  step = 10,
  f.plot.polygon = F
)
```

Arguments

`centre` An array of two coordinates in UTM 33 (epsg:25833)

`dem` A SpatRaster of a digital elevation/surface model in UTM 33 (epsg:25833)

`level` A height above the ground of the sensor in metres, level 0 is set to 2 metres

`step` An interval in degrees at which the horizon will be computed, default is every 10 deg.

`f.plot.polygon` A boolean setting boundary values to -20 deg to facilitate plotting as polygon

Value

A dataframe with ‘horizon_height‘ in degrees and ‘azimuth‘ angle in degrees at which the horizon is computed in degrees

References

https://grass.osgeo.org/grass83/manuals/r.horizon.html

Examples

```
# Load the station metadata including location and level
stn <- get_latlon_frost(stationid,paramid)
stn.id      <- stn$id.stationid
stn.centre  <- stn  %>% st_coordinates
stn.level   <- stn$id.level

# Load a digital elevation model
dsm   <- download_dem_kartverket(stn.id,stn.centre,name="dom",dx=100,resx=1)

# Compute the horizon
```

```
compute_horizon(stn.centre,dem)
compute_horizon(stn.centre,dem,level=stn.level,step=.01,f.plot.polygon=T)
```

compute_skyviewfactor *Sky View Factor (SVF)*

Description

Compute Sky View Factor from horizon data

Usage

```
compute_skyviewfactor(horizon = NULL)
```

Arguments

horizon	An array of horizon height in degrees from "compute_horizon"

Value

Estimated sky view factor, 1 defines an open sky view and 0 a totally obstructed sky view

References

https://github.com/OSGeo/grass-addons/blob/grass8/src/raster/r.skyview/r.skyview.py

Examples

```
compute_skyviewfactor(horizon)
```

compute_sun_position *Sun position in the sky*

Description

Compute sun position in the sky from a station location in UTM 33

Usage

```
compute_sun_position(stn = NULL, f.hour = F)
```

Arguments

`stn`	A SpatVector with station attributes from `"get_latlon_frost"`
`f.hour`	A boolean value to compute sun position for hours if TRUE and for days if FALSE

Value

dataframe with inclination of the sun position in the sky in degrees, azimuth in degrees and timestamp as factor

References

https://github.com/adokter/suntools/

Examples

```
compute_sun_position(stn, f.hour = TRUE)
compute_sun_position(stn, f.hour = FALSE)
```

download_dem_kartverket

Download digital elevation models from Kartverket's WCS API

Description

Define a GetCapabilities request URL using OSW4R and Kartverket's Web Coverage Service that downloads a DEM from a bounding box and a DEM type (i.e. "dom" or "dtm"). The bounding box is centered to a parsed location and a parsed radius set the extent. The downloaded DEM is a SpatRaster object. If the DEM file already exists, it is loaded by default unless f.overwrite is set to TRUE

Usage

```
download_dem_kartverket(
  stationid = NULL,
  centre = NULL,
  name = "dom",
  dx = 100,
  resx = dx/100,
  path = "data/dem",
  f.overwrite = FALSE
)
```

Arguments

`stationid` A station ID used for the DEM file name

`centre` A coordinate array (i.e. ‘c(x, y)‘) of the station in UTM 33 (i.e. epsg:25833)

`name` A name of the DEM to download, either "dtm" a terrain model or the default "dom" a surface model

`dx` A distance in metre or radius defining the extent of the bounding box from the centre point, default ‘100‘ metres

`resx` A horizontal resolution in metre, default is ‘dx/100‘ if greater than ‘1‘ metre

`path` A directory path defining where will be saved the data, default path is "data/dem"

`f.overwrite` A boolean whether the DEM file should be overwritten, default ‘FALSE‘

Value

A Digital Elevation Model

References

https://kartkatalog.geonorge.no/metadata/nasjonal-hoeydemodell-digital-terrengmodell-25833-wcs/0f0a0f38-00c4-4213-a9e5-2d861dc4abb0

https://kartkatalog.geonorge.no/metadata/nasjonal-hoeydemodell-digital-overflatemodell-25833-wcs/e36ea427-13a1-4d7c-be82-977068dfc3e3

https://cran.r-project.org/web/packages/ows4R/vignettes/wcs.html

Examples

```
# Load data
stationid <- 18700
centre <- stn  %>% st_coordinates
path   <- "data/dem"
dem    <- download_dem_kartverket(stationid,centre,name="dtm",dx=100,resx=1,path=path)
dsm    <- download_dem_kartverket(stationid,centre,name="dom",dx=100,resx=1,path=path)
demkm  <- download_dem_kartverket(stationid,centre,name="dtm",dx=20e3,resx=20,path=path)
```

`download_dem_kartverket_ogc`

Download DEM from Kartverket’s NEWEST OGC API

Description

Download DEM from Kartverket’s NEWEST OGC API

Usage

```
download_dem_kartverket_ogc(
  stationid = 18703,
  centre,
  name = "dom",
  dx = 100,
  resx = 1,
  f.OGC = T
)
```

Arguments

`stationid` A station ID used for the DEM file name

`centre` A coordinate array (i.e. ‘c(x, y)‘) of the station in UTM 33 (i.e. ‘epsg:25833‘)

`name` A name of the DEM to download, either ‘"dtm"‘ a terrain model or the default ‘"dom"‘ a surface model

`dx` A distance in metre or radius defining the extent of the bounding box from the centre point, default ‘100‘ metres

`resx` A horizontal resolution in metre, default is ‘dx/100‘ if greater than ‘1‘ metre

`f.OGC` A boolean default is ‘TRUE‘

Value

A DEM

Examples

```
download_dem_kartverket_ogc()
```

get_latlon_frost *Get station and sensor metadata from Frost API*

Description

Fetch station metadata based on station number and parameter id from Frost v1. The function gets station name, location and sensor details such as level, exposure and performance

Usage

```
get_latlon_frost(stationid = 18700, paramid = NULL)
```

Arguments

`stationid` A station number as integer and defined by met.no, default is 18700 - Blindern

`paramid` A parameter number as integer and defined by met.no, default is NULL, thus getting all paramid available

Value

Station metadata

References

https://frost-beta.met.no/docs/codeexamples

Examples

```
get_latlon_frost(stationid=18700)
get_latlon_frost(stationid=18700,paramid=211)
```

`get_tile_wms` *Get Web Mapping Service data tiles*

Description

Fetch map tiles from publicly available WMS released by Nibio

Usage

```
get_tile_wms(box = NULL, layer = "CORINE_Land_Cover_2012", px = 500)
```

Arguments

`box` A SpatExtent defining the area to plot

`layer` A name of a layer provided by WMS such as "ar5", "CORINE_Land_Cover_2012" (default) and "Urban_Atlas_Lu_Lc_2012"

`px` A pixel number that defines the resolution of the image/tile, default is 500 px

Value

A map tile

References

https://nibio.no/tjenester/wms-tjenester

Examples

```
tile <- get_tile_wms(box, layer = "ar5")
tile <- get_tile_wms(box, layer = "CORINE_Land_Cover_2012" )
tile <- get_tile_wms(box, layer = "Urban_Atlas_Lu_Lc_2012" )
```

`load_data_ar5` *Load FKB-AR5 from vector files stored locally (restricted access to download)*

Description

Load vector FKB-AR5 files to assess area type aka land cover, alternatively get WMS tile (deprecated, use "get_tile_wms")

Usage

```
load_data_ar5(box = NULL, f.wms = TRUE, layer = "Arealtype", px = 500)
```

Arguments

box	A SpatExtent defining the area to plot
f.wms	A boolean to switch to WMS image tile instead of loading file, default TRUE (deprecated)
layer	A name of a layer provided by WMS. For "ar5", it is "Arealtype" as default
px	A pixel number that defines the resolution of the image/tile, default is 500 px

Value

A map tile

References

https://kartkatalog.geonorge.no/metadata/fkb-ar5/166382b4-82d6-4ea9-a68e-6fd0c87bf788

Examples

```
ar5 <- load_data_ar5(box,f.wms=F)
```

`plot_dem_rayshader` *Plot digital surface/elevation model in 3D with shadow rendering*

Description

Use rayshader to plot surface elevation model near a weather station from four cardinal angles

Usage

```
plot_dem_rayshader(stn = NULL, dsm = NULL, path = NULL)
```

Arguments

`stn`	A SpatVector with station attributes from "`get_latlon_frost`"
`dsm`	A SpatRaster of a digital surface model around the station, expected radius is 100 m
`path`	A string path that defines where to save the plot, if NULL (default) the plot is printed on-screen and not saved

Value

A rendered image

References

https://www.rayshader.com/

Examples

```
plot_dem_rayshader(stn,dsm, path='plot/dem3D')
```

`plot_station_horizon_sun`

Plot horizon with sun positions for a given station location

Description

Fetch station metadata based on station number and parameter id from Frost v1. The function gets station name, location and sensor details such as level, exposure and performance

Usage

```
plot_station_horizon_sun(
  stn = NULL,
  dem = NULL,
  dsm = NULL,
  demkm = NULL,
  path = NULL
)
```

Arguments

`stn`	A SpatVector with station attributes from "`get_latlon_frost`"
`dem`	A SpatRaster of a digital elevation/terrain model around the station, expected radius is 100 m
`dsm`	A SpatRaster of a digital surface model around the station, expected radius is 100 m

demkm	A SpatRaster of a digital elevation/terrain model around the station, expected radius is 20 km
path	A directory path defining where will be saved the plots, if path is NULL the plots are printed to the console

Value

Sun diagram with station metadata

References

https://frost-beta.met.no/docs/codeexamples

Examples

```
# Load data
stn <- get_latlon_frost(stationid=18700)
dem   <- download_dem_kartverket(stationid,centre,name="dtm",dx=100,resx=1)
dsm   <- download_dem_kartverket(stationid,centre,name="dom",dx=100,resx=1)
demkm <- download_dem_kartverket(stationid,centre,name="dtm",dx=20e3,resx=20)
path <- sprintf("station_location_files/output/%i",stn$id.stationid)
# path <- 'plot/horizon'
# Plot sun diagram
plot_station_horizon_sun(stn, dem, dsm, demkm, path=path)
```

plot_station_siting_context

Weather station context

Description

Plots weather station’s sun diagram and background maps to assess the exposure of a station and to compute its WMO/met.no siting classification

Usage

```
plot_station_siting_context(
  stationid = 18700,
  paramid = 211,
  f.verbose = TRUE,
  f.pdf = FALSE
)
```

Arguments

stationid	A station number as integer and defined by met.no
paramid	A parameter number as integer and defined by met.no
f.verbose	A boolean string to print debug messages, default is TRUE
f.pdf	A boolean string to combine all plots into a pdf file, default is FALSE

Value

None

References

https://community.wmo.int/en/activity-areas/imop/siting-classification#MembersTools

Examples

```
# Plot sun diagram and map infos for a weather station
plot_station_siting_context(stationid=18700)
plot_station_siting_context(stationid=18700,paramid=211,f.verbose=T)
```

---

plot_tile_station *Plot station location with background WMS tiles*

Description

Plots publicly-available maps, atlas, land cover or satellite imagery near a weather station based on maptiles and custom-made "get_tile_wms"

Usage

```
plot_tile_station(
  stn = NULL,
  box = NULL,
  tile_name = "osm",
  dsm = NULL,
  path = NULL
)
```

Arguments

`stn` A SpatVector with station attributes from "get_latlon_frost"

`box` A SpatExtent defining the area to plot

`tile_name` A string defining the type of tile to plot among "osm" (map, default), "esri" (satellite imagery), "ar5" (area type), "clc" (Corine land cover) and "urban" (urban atlas)

`dsm` A SpatRaster of a digital surface model around the station, expected radius is 100 m

`path` A string path that defines where to save the plot, if NULL (default) the plot is printed on-screen and not saved

Value

A ggplot object

References

https://github.com/riatelab/maptiles

Examples

```
g <- plot_tile_station(stn, box, tile_name = "esri")
g
plot_tile_station(stn,box,tile_name="esri", path=path)
plot_tile_station(stn,box,tile_name="ar5", path=path)
plot_tile_station(stn,box,tile_name="clc", path=path)
plot_tile_station(stn,box,tile_name="urban", path=path)
plot_tile_station(stn,box,tile_name="osm",dsm=dsm, path="plot/map")
```

Index