

Package: LCZ4r (via r-universe)

October 29, 2026

Type Package

Title Tools for Local Climate Zones and Urban Heat Island Analysis

Version 0.1.0

Maintainer Max Anjos <maxanjos@campus.ul.pt>

Description LCZ4r is a comprehensive R package designed for analyzing and visualizing Local Climate Zones (LCZ) and Urban Heat Islands (UHI) at various spatial and temporal resolutions. It provides a suite of functions to work with LCZ data, perform urban climate analysis, and generate insights into temperature variations within urban areas.

License MIT + file LICENSE

URL <https://bymaxanjos.github.io/LCZ4r/>

BugReports <https://github.com/ByMaxAnjos/LCZ4r/issues>

Depends R (>= 3.2.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports dplyr (>= 1.1.2), tidyterra (>= 0.3.1), magrittr (>= 2.0.3),
osmdata (>= 0.2.5), raster (>= 3.5.15), rlang (>= 1.1.1), sf
(>= 1.0.7), stars (>= 0.5.5), terra (>= 1.5.21), ggplot2 (>= 3.5.1), scales (>= 1.3.0), tibble (>= 3.2.1), openair (>= 2.9.0), recipes (>= 1.0.9), automap (>= 1.1.9), gstat (>= 2.1.0), MetBrewer (>= 0.2.0), lubridate (>= 1.9.2)

Suggests knitr, tidyr, data.table, openssl, httr2, curl, rmarkdown,
htmltools, htmlwidgets, covr

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository <https://bymaxanjos.r-universe.dev>

Date/Publication 2026-10-29 00:30:39 UTC
RemoteUrl <https://github.com/ByMaxAnjos/LCZ4r>
RemoteRef HEAD
RemoteSha 2c3a6b318d01ef79edc9b7624ce764291b04cea0

Contents

lcz_anomaly	2
lcz_anomaly_map	5
lcz_cal_area	8
lcz_data	9
lcz_get_map	10
lcz_get_map_euro	11
lcz_get_map_generator	12
lcz_get_map_usa	13
lcz_get_map2	14
lcz_get_parameters	15
lcz_interp_eval	17
lcz_interp_map	20
lcz_plot_interp	23
lcz_plot_map	24
lcz_plot_parameters	25
lcz_ts	27
lcz_tutorial	30
lcz_uhi_intensity	30
Index	34

lcz_anomaly	<i>Explore LCZ Thermal Anomalies</i>
-------------	--------------------------------------

Description

This function generates a graphical representation of thermal anomaly for different Local Climate Zones (LCZs). More details: https://bymaxanjos.github.io/LCZ4r/articles/Introd_local_LCZ4r.html

Usage

```
lcz_anomaly(  
  x,  
  data_frame = "",  
  var = "",  
  station_id = "",  
  ...,  
  time.freq = "hour",
```

```

extract.method = "simple",
plot_type = "diverging_bar",
by = NULL,
impute = NULL,
iplot = TRUE,
isave = FALSE,
save_extension = "png",
inclusive = FALSE,
palette = "Okeeffe1",
ylab = "Air temperature anomaly",
xlab = "Stations",
title = "",
caption = "",
legend_name = "Anomaly"
)

```

Arguments

<code>x</code>	A <code>SpatRaster</code> object containing the LCZ map. The LCZ map can be obtained using the <code>lcz_get_map()</code> function.
<code>data_frame</code>	A data frame containing air temperature measurements and station IDs. The data frame should have a date field in hourly or higher resolution format.
<code>var</code>	The name of the variable in the data frame representing air temperature.
<code>station_id</code>	The name of the variable in the data frame representing station IDs.
<code>...</code>	Additional arguments for the <code>selectByDate</code> function from the <code>openair</code> package. These arguments allow for flexible selection of specific time periods (year, month, day, hour). Examples of how to use these arguments include: <ul style="list-style-type: none"> • Year(s): Numeric value(s) specifying the year(s) to select. For example, <code>year = 1998:2004</code> selects all years between 1998 and 2004 (inclusive), while <code>year = c(1998, 2004)</code> selects only the years 1998 and 2004. • Month(s): Numeric or character value(s) specifying the months to select. Numeric examples: <code>month = 1:6</code> (January to June), or character examples: <code>month = c("January", "December")</code>. • Day(s): Numeric value(s) specifying the days to select. For instance, <code>day = 1:30</code> selects days from 1 to 30, or <code>day = 15</code> selects only the 15th day of the month. • Hour(s): Numeric value(s) specifying the hours to select. For example, <code>hour = 0:23</code> selects all hours in a day, while <code>hour = 9</code> selects only the 9th hour. • Start date: A string specifying the start date in either <code>start="DD/MM/YYYY"</code> (e.g., "1/2/1999") or <code>"YYYY-mm-dd"</code> format (e.g., "1999-02-01"). • End date: A string specifying the end date in either <code>end="DD/MM/YYYY"</code> (e.g., "1/2/1999") or <code>"YYYY-mm-dd"</code> format (e.g., "1999-02-01").
<code>time.freq</code>	Defines the time period to average to. Default is "hour", but includes "day", "week", "month" or "year".

extract.method	<p>A character string specifying the method used to assign the LCZ class to each station point. The default is "simple". Available methods are:</p> <ul style="list-style-type: none"> • simple: Assigns the LCZ class based on the value of the raster cell in which the point falls. It often is used in low-density observational network. • two.step: Assigns LCZs to stations while filtering out those located in heterogeneous LCZ areas. This method requires that at least 80% of the pixels within a 5×5 kernel match the LCZ of the center pixel (Daniel et al., 2017). Note that this method reduces the number of stations. It often is used in ultra and high-density observational network, especially in LCZ classes with multiple stations. • bilinear: Interpolates the LCZ class values from the four nearest raster cells surrounding the point.
plot_type	<p>A character string indicating the type of plot to generate. Options include:</p> <ul style="list-style-type: none"> • diverging_bar: A horizontal bar plot that diverges from the center (zero), with positive anomalies extending to the right and negative anomalies to the left. This plot is good for showing the extent and direction of anomalies in a compact format. • bar: A bar plot showing the magnitude of the anomaly for each station, colored by whether the anomaly is positive or negative. This plot is good for comparing anomalies across stations. • dot: A dot plot that displays both the mean temperature values and the reference values, with lines connecting them. The size or color of the dots can indicate the magnitude of the anomaly. Ideal for showing both absolute temperature values and their anomalies. • lollipop: A lollipop plot where each "stick" represents an anomaly value and the dots at the top represent the size of the anomaly. Useful for clearly showing positive and negative anomalies in a minimalist way.
by	<p>data frame time-series split: "year", "season", "seasonyear", "month", "monthyear", "weekday", "weekend", "site", "daylight", "dst" (daylight saving time). Note that the Daylight option may result in both daytime and nighttime hours being represented in UTC. See NOAA https://gml.noaa.gov/grad/solcalc/ and argument <i>type</i> in openair package: https://openair-project.github.io/book/sections/intro/openair-package.html#the-type-option</p>
impute	Method to impute missing values ("mean", "median", "knn", "bag").
iplot	Set to TRUE to return a plot. If FALSE, a data frame is returned.
isave	Set to TRUE to save all results (plot, time-series) into your directory.
save_extension	A character string indicating the file format for saving the plot. Options include: "png", "jpg", "jpeg", "tif", "pdf", "svg". The default is "png".
inclusive	Set to TRUE to a colorblind-friendly palette.
palette	A character string specifying the color palette to use. The default is "OKeeffe1". You can choose from palettes available in MetBrewer: https://github.com/BlakeRMills/MetBrewer?tab=readme-ov-file#palettes .
ylab	y-axis name.
xlab	y-axis name. Default is "Station"

title	y-axis name. Default is "" "".
caption	source data. Default can be "Source: LCZ4r, Stewart and Oke, 2012; Demuzere et al.2022."".
legend_name	Legend name for dot plot. Default is "Anomaly".

Value

A visual representation of the anomalies of air temperature of LCZ in ggplot or data frame .csv format.

Author(s)

Max Anjos (<https://github.com/ByMaxAnjos>)

See Also

See the documentation for lcz_get_map() to obtain an LCZ map.

Examples

```
## Not run:
# Hourly air temperature values in 2019.
my_res <- lcz_anomaly(lcz_map, data_frame = lcz_data, var = "airT",
                     station_id = "station", year = 2020)

## End(Not run)
```

lcz_anomaly_map	<i>Interpolate spatial and temporal thermal anomaly using a Local Climate Zones (LCZ).</i>
-----------------	--

Description

Generates an interpolated spatiotemporal anomaly map using LCZ as background. This function applies geostatistical interpolation techniques (e.g., kriging) to estimate anomaly values (e.g., air temperature) across the study area, integrating LCZ classifications. The function allows for flexible time period selection, like hour, daytime, nighttime and so on.

Usage

```
lcz_anomaly_map(
  x,
  data_frame = "",
  var = "",
  station_id = "",
  ...,
  sp.res = 100,
  tp.res = "hour",
```

```

vg.model = "Sph",
extract.method = "simple",
by = NULL,
impute = NULL,
isave = FALSE,
LCZinterp = TRUE
)

```

Arguments

x	A SpatRaster object representing the LCZ map, obtained via <code>lcz_get_map()</code> functions.
data_frame	A data frame containing air temperature measurements with station metadata. The data must include a date column in hourly or higher resolution.
var	Character. The variable to interpolate (e.g., "airT" for air temperature).
station_id	Character. The column name in data_frame identifying weather stations.
...	Additional arguments passed to <code>selectByDate</code> from the <code>openair</code> package for flexible time selection. Options include: <ul style="list-style-type: none"> • Year(s): <code>year = 1998:2004</code> (selects 1998 to 2004) or <code>year = c(1998, 2004)</code>. • Month(s): <code>month = 1:6</code> (January to June) or <code>month = c("January", "December")</code>. • Day(s): <code>day = 1:30</code> (days 1 to 30) or <code>day = 15</code>. • Hour(s): <code>hour = 0:23</code> (all hours) or <code>hour = 9</code>. • Start/End Date: "YYYY-mm-dd" format (e.g., "1999-02-01").
sp.res	Numeric. Spatial resolution in meters (default: 100).
tp.res	Character. Temporal resolution for averaging ("hour", "day", etc.).
vg.model	Character. Variogram model for kriging. Default is "Sph" (spherical). Available models: <ul style="list-style-type: none"> • "Sph": Spherical model. • "Exp": Exponential model. • "Gau": Gaussian model. • "Ste": M. Stein's parameterization.
extract.method	Character. The method used to assign LCZ classes to stations: <ul style="list-style-type: none"> • "simple": Assigns the LCZ based on the raster cell where the station is located. It often is used in low-density observational network. • "two.step": Filters out stations in heterogeneous LCZ areas (Daniel et al., 2017). It often is used in ultra and high-density observational network, especially in LCZ classes with multiple stations. • "bilinear": Uses bilinear interpolation to assign LCZ values.
by	data frame time-series split: "year", "season", "seasonyear", "month", "monthyear", "weekday", "weekend", "site", "daylight", "dst" (daylight saving time). Note that the Daylight option may result in both daytime and nighttime hours being represented in UTC . See NOAA https://gml.noaa.gov/grad/solcalc/ and argument <i>type</i> in <code>openair</code> package: https://openair-project.github.io/book/sections/intro/openair-package.html#the-type-option

impute	Character. Method to impute missing values ("mean", "median", "knn", "bag").
isave	Logical. If TRUE, saves the plot to the working directory.
LCZinterp	Logical. If TRUE (default), applies LCZ-based interpolation.

Value

A spatially interpolated map in terra raster GeoTIFF format.

Author(s)

Max Anjos (<https://github.com/ByMaxAnjos>)

References

Anjos, M., Targino, A. C., Krecl, P., Oukawa, G. Y. & Braga, R. F. Analysis of the urban heat island under different synoptic patterns using local climate zones. *Build. Environ.* 185, 107268 (2020).
 Fenner, D., Meier, F., Bechtel, B., Otto, M. & Scherer, D. (2017). Intra and inter 'local climate zone' variability of air temperature as observed by crowdsourced citizen weather stations in Berlin, Germany. *Meteorol. Z.*, 26, 525–547. <http://www.gstat.org/>

See Also

See `lcz_anomaly()` for LCZ interpolation evaluation.

Examples

```
## Not run:
# Load sample data and LCZ map
data("lcz_data")
lcz_map <- lcz_get_map_generator(ID = "8576bde60bfe774e335190f2e8fdd125dd9f4299")

# Plot the LCZ map
lcz_plot_map(lcz_map)

# Perform interpolation anomalies for 2020-01-02 at 04:00h
my_interp <- lcz_anomaly_map(
  lcz_map, data_frame = lcz_data, var = "airT",
  station_id = "station", year = 2020, month = 1, day = 2, hour = 4,
  sp.res = 100, tp.res = "hour", vg.model = "Sph",
  LCZinterp = TRUE, isave = TRUE
)

# Perform interpolation for an entire day (returns a raster stack of 24 hours)
my_interp <- lcz_anomaly_map(
  lcz_map, data_frame = lcz_data, var = "airT",
  station_id = "station", year = 2020, month = 1, day = 2,
  sp.res = 100, tp.res = "hour", vg.model = "Sph",
  LCZinterp = TRUE, isave = TRUE
)

## End(Not run)
```

lcz_cal_area	<i>Calculate LCZ areas</i>
--------------	----------------------------

Description

This function calculates the areas of LCZ classes in both percentage and square kilometers.

Usage

```
lcz_cal_area(
  x,
  plot_type = "bar",
  iplot = TRUE,
  isave = FALSE,
  save_extension = "png",
  show_legend = TRUE,
  inclusive = FALSE,
  xlab = "LCZ code",
  ylab = "Area [km2]",
  ...
)
```

Arguments

<code>x</code>	A raster <code>SpatRaster</code> layer containing LCZ classes.
<code>plot_type</code>	Type of plot to generate: "bar", "pie", or "donut". Default is "bar".
<code>iplot</code>	Logical, indicating whether to create a plot (default is TRUE).
<code>isave</code>	Logical, whether to save the plot to your directory (default is FALSE).
<code>save_extension</code>	File format for saving the plot. Options: "png", "jpg", "jpeg", "tif", "pdf", "svg" (default is "png").
<code>show_legend</code>	Logical. If TRUE, displays the legend on the plot. If FALSE, the legend will be hidden. Default is TRUE.
<code>inclusive</code>	Logical. If TRUE, use a colorblind-friendly palette. Default is FALSE.
<code>xlab</code>	Character, label for the x-axis (default is "LCZ code").
<code>ylab</code>	Character, label for the y-axis (default is "Area (km2)").
<code>...</code>	Additional arguments to modify axis, legend, and plot labels, including title, subtitle, and caption.

Value

A summary table of LCZ class areas if `plot = FALSE`, otherwise, the specified plot in ggplot format.

See Also

See the documentation for `lcz_get_map()` to obtain an LCZ map.

Examples

```
## Not run:
# Calculate LCZ area and visualize as bar plot
lcz_cal_area(my_lcz_map, plot_type = "bar")

# Visualize as pie chart
lcz_cal_area(my_lcz_map, plot_type = "pie")

# Save donut chart plot
lcz_cal_area(my_lcz_map, plot_type = "donut", isave = TRUE)

# Return a dataframe
lcz_cal_area(my_lcz_map, plot_type = "donut", iplot = FALSE)

## End(Not run)
```

lcz_data

*Air Temperature LCZ Data***Description**

This dataset is a subset of data from the Urban Climate Observatory (UCO) in Berlin, Germany. It contains hourly air temperature readings from 23 meteorological stations for the period of 2019-2020.

Usage

```
lcz_data
```

Format

A data frame with 386,354 rows and 5 columns:

date Date and time in POSIXct format: "2019-01-01 00:00:00"

station Name of the meteorological station

airT Air temperature in degree celsius

Latitude Latitude of the station

Longitude Longitude of the station

Source

[UCO Berlin](#)

Examples

```
data(lcz_data)
```

lcz_get_map

*Download the LCZ map***Description**

This function retrieves the Local Climate Zone (LCZ) global mapping dataset. It allows you to obtain the LCZ map for a specific area of interest, which can be a city, state, region, or custom-defined shape.

Usage

```
lcz_get_map(city = NULL, roi = NULL, isave_map = FALSE, isave_global = FALSE)
```

Arguments

city	A character string specifying the name of your target area based on the Open-StreetMap project.
roi	Optionally, you can provide a Region of Interest (ROI) in ESRI shapefile format to clip the LCZ map to a custom area.
isave_map	Logical. Set to TRUE if you wish to save the resulting clipped map as a raster TIFF file on your local machine.
isave_global	Logical. Set to TRUE if you wish to save the global LCZ map as a raster TIFF file on your local machine.

Value

A Terra raster TIFF file containing LCZ classes (1-17).

References

Demuzere, M., Kittner, J., Martilli, A., Mills, G., Moede, C., Stewart, I. D., van Vliet, J., and Bechtel, B. (2022). A global map of Local Climate Zones to support earth system modelling and urban scale environmental science, *Earth Syst. Sci. Data* 14(8) 3835-3873. DOI:<https://doi.org/10.5194/essd-14-3835-2022> Zenodo. <https://doi.org/10.5281/zenodo.6364594> Stewart ID, Oke TR. (2012). Local Climate Zones for Urban Temperature Studies. *Bull Am Meteorol Soc.* 93(12):1879-1900. doi:10.1175/BAMS-D-11-00019.1

Examples

```
## Not run:
# Load the LCZ map for a city
my_lcz_city <- lcz_get_map(city = "your city")

# Get LCZ map for a custom region of interest
custom_roi <- sf::st_read("custom_roi.shp")
roi_lcz <- lcz_get_map(roi = custom_roi, isave_map = TRUE)
```

```
# Retrieve the LCZ map for a country
my_lcz_country <- lcz_get_map(city = "Brazil")

## End(Not run)
```

lcz_get_map_euro

Download LCZ map from the European

Description

This function retrieves the Local Climate Zone (LCZ) European mapping dataset created by Demuzere et al. (2019) and available at <https://doi.org/10.1371/journal.pone.0214474>. It allows you to obtain the LCZ map for a specific area of interest, which can be a city, state, region, or custom-defined shape.

Usage

```
lcz_get_map_euro(
  city = NULL,
  roi = NULL,
  isave_map = FALSE,
  isave_euro = FALSE
)
```

Arguments

city	A character string specifying the name of your target area based on the OpenStreetMap project.
roi	Optionally, you can provide a Region of Interest (ROI) in ESRI shapefile format to clip the LCZ map to a custom area.
isave_map	Logical. Set to TRUE if you wish to save the resulting clipped map as a raster TIFF file on your local machine.
isave_euro	Logical. Set to TRUE if you wish to save the European LCZ map as a raster TIFF file on your local machine.

Value

A Terra raster TIFF file containing LCZ classes (1-17).

References

Demuzere, M., Bechtel, B., Middel, A., & Mills, G. (2019). Mapping Europe into local climate zones. PLOS ONE, 14(4), e0214474. DOI:<https://doi.org/10.1371/journal.pone.0214474> Demuzere, M.; Bechtel, B.; Middel, A.; Mills, G. (2019): European LCZ map. Figshare. Dataset. <https://doi.org/10.6084/m9.figshare.13322450> Anjos, M. (2024). LCZ4r Package Dataset. Zenodo. <https://doi.org/10.5281/zenodo.10835692>

Examples

```
## Not run:
# Example 1: Load the LCZ map for a city
my_lcz_city <- lcz_get_map_euro(city = "Berlin")

# Example 2: Get LCZ map for a custom region of interest
custom_roi <- sf::st_read("custom_roi.shp")
roi_lcz <- lcz_get_map_euro(roi = custom_roi)

## End(Not run)
```

lcz_get_map_generator *Download the LCZ map from LCZ Generator*

Description

This function retrieves the LCZ map from LCZ Generator platform

Usage

```
lcz_get_map_generator(
  ID = "3110e623fbe4e73b1cde55f0e9832c4f5640ac21",
  band = "lczFilter",
  isave_map = FALSE
)
```

Arguments

ID	A code specifying the ID generated by LCZ Factsheet (e.g, "3110e623fbe4e73b1cde55f0e9832c4f5640ac21"). More details: https://lcz-generator.rub.de/factsheets/3110e623fbe4e73b1cde55f0e9832c4f5640ac21_factsheet.html
band	Select the feature to use. The available options are: "lcz" and "lczFilter". The default is "lczFilter".
isave_map	Logical. Set to TRUE if you wish to save the resulting map as a raster TIFF file on your local machine.

Value

A Terra raster TIFF file containing LCZ classes (1-17).

References

Demuzere, M., Kittner, J., Bechtel, B. (2021). LCZ Generator: a web application to create Local Climate Zone maps. *Frontiers in Environmental Science* 9:637455. <https://doi.org/10.3389/fenvs.2021.637455>

Examples

```
## Not run:
# Load the Rio de Janeiro map from LCZ Generator
my_lcz_map <- lcz_get_map_generator(ID = "3110e623fbe4e73b1cde55f0e9832c4f5640ac21")

## End(Not run)
```

lcz_get_map_usa	<i>Download your LCZ map from the Continental United States</i>
-----------------	---

Description

This function retrieves the Local Climate Zone (LCZ) CONUS LCZ mapping dataset created by Demuzere et al. (2020) and available at <https://doi.org/10.1038/s41597-020-00605-z>. It allows you to obtain the LCZ map for a specific area of interest, which can be a city, state, region, or custom-defined shape.

Usage

```
lcz_get_map_usa(city = NULL, roi = NULL, isave_map = FALSE, isave_usa = FALSE)
```

Arguments

city	A character string specifying the name of your target area based on the OpenStreetMap project.
roi	Optionally, you can provide a Region of Interest (ROI) in ESRI shapefile format to clip the LCZ map to a custom area.
isave_map	Logical. Set to TRUE if you wish to save the resulting clipped map as a raster TIFF file on your local machine.
isave_usa	Logical. Set to TRUE if you wish to save the USA LCZ map as a raster TIFF file on your local machine.

Value

A Terra raster TIFF file containing LCZ classes (1-17).

References

Demuzere, M., Hankey, S., Mills, G., Zhang, W., Lu, T., & Bechtel, B. (2020). Combining expert and crowd-sourced training data to map urban form and functions for the continental US. *Scientific Data*, 7(1), 264. DOI:<https://doi.org/10.1038/s41597-020-00605-z>

Demuzere, M., Hankey, S., Mills, G., Zhang, W., Lu, T., Bechtel B. (2020): CONUS-wide LCZ map and Training Areas. Figshare. Dataset. <https://doi.org/10.6084/m9.figshare.11416950>

Anjos, M. (2024). LCZ4r Package Dataset. Zenodo. <https://doi.org/10.5281/zenodo.10835692>

Examples

```
## Not run:
# Example 1: Load the LCZ map for a city
my_lcz_city <- lcz_get_map_usa(city = "New York")

# Example 2: Get LCZ map for a custom region of interest
custom_roi <- sf::st_read("custom_roi.shp")
roi_lcz <- lcz_get_map_usa(roi = custom_roi)

## End(Not run)
```

lcz_get_map2	<i>Obtain the LCZ map from LCZ generator or with downloaded global LCZ map</i>
--------------	--

Description

This function retrieves the Local Climate Zone (LCZ) global mapping dataset created by Demuzere et al. (2022) and available at <https://doi.org/10.5194/essd-14-3835-2022>. It allows you to obtain the LCZ map for a specific area of interest, which can be a city, state, region, or custom-defined shape.

Usage

```
lcz_get_map2(x, city = NULL, roi = NULL, isave_map = FALSE)
```

Arguments

x	A SpatRaster object containing the LCZ map. It can be download using these links: https://lcz-generator.rub.de/ , https://zenodo.org/records/8419340/files/lcz_filter_v3.tif?download=1 . Then, it suggested use <code>terra::rast(path/name.tif)</code> to import the raster .tif into R.
city	A character string specifying the name of your target area based on the OpenStreetMap project.
roi	Optionally, you can provide a Region of Interest (ROI) in ESRI shapefile format to clip the LCZ map to a custom area.
isave_map	Logical. Set to TRUE if you wish to save the resulting clipped map as a raster TIFF file on your local machine.

Value

A Terra raster TIFF file containing LCZ classes for the specified area of interest.

Examples

```
## Not run:
# Go to https://lcz-generator.rub.de/
my_map <- terra::rast("path/name.tif")
my_lcz_map <- lcz_get_map2(my_map, city = "Name")

# Load the LCZ map from your PC
my_map <- terra::rast("path/lcz_filter_v3.tif")

# Get LCZ map for a custom region of interest
custom_roi <- sf::st_read("custom_roi.shp")
roi_lcz <- lcz_get_map2(my_map, roi = custom_roi)

# Retrieve the LCZ map for a country (no custom ROI specified)
my_lcz_country <- lcz_get_map2(my_map, city = "Lisbon")

## End(Not run)
```

lcz_get_parameters	<i>Retrieve urban canopy LCZ parameters</i>
--------------------	---

Description

This function extracts 34 LCZ parameters based on the classification scheme developed by Stewart and Oke (2012). LCZs provide valuable information about urban morphology and are useful for urban climate and environmental studies.

Usage

```
lcz_get_parameters(x, iselect = "", istack = TRUE, ishp = FALSE, isave = FALSE)
```

Arguments

x	A LCZ map in .tiff format SpatRaster object containing LCZ classes.
iselect	Code vector. Specify one or more parameter: <ul style="list-style-type: none"> • SVFmean: Mean Sky View Factor • SVFmax: Maximum Sky View Factor • SVFmin: Minimum Sky View Factor • ARmean: Mean Aspect Ratio • ARmax: Maximum Aspect Ratio • ARmin: Minimum Aspect Ratio • BSFmean: Mean Building Surface Fraction • BSFmax: Maximum Building Surface Fraction • BSFmin: Minimum Building Surface Fraction • ISFmean: Mean Impervious Surface Fraction

- **ISFmax**: Maximum Impervious Surface Fraction
- **ISFmin**: Minimum Impervious Surface Fraction
- **PSFmean**: Mean Vegetation Surface Fraction
- **PSFmax**: Maximum Vegetation Surface Fraction
- **PSFmin**: Minimum Vegetation Surface Fraction
- **TSFmean**: Mean Tree Surface Fraction
- **TSFmax**: Maximum Tree Surface Fraction
- **TSFmin**: Minimum Tree Surface Fraction
- **HREmean**: Mean Height Roughness Elements
- **HREmax**: Maximum Height Roughness Elements
- **HREmin**: Minimum Height Roughness Elements
- **TRCmean**: Mean Terrain Roughness class
- **TRCmax**: Maximum Terrain Roughness class
- **TRCmin**: Minimum Terrain Roughness class
- **SADmean**: Mean Surface Admittance
- **SADmax**: Maximum Surface Admittance
- **SADmin**: Minimum Surface Admittance
- **SALmean**: Mean Surface Albedo
- **SALmax**: Maximum Surface Albedo
- **SALmin**: Minimum Surface Albedo
- **AHmean**: Mean Anthropogenic Heat Outupt
- **AHmax**: Maximum Anthropogenic Heat Outupt
- **AHmin**: Minimum Anthropogenic Heat Outupt
- **z0**: Roughness Lenght class

istack	Logical. If TRUE, returns all parameters as a raster stack. If FALSE, returns a list of individual parameter rasters.
ishp	Logical. If TRUE, returns all parameters as an ESRI shapefile.
isave	Set to TRUE if you want to save the your clipped resulting map as a raster.tiff or .shp file on your local machine.

Value

If 'istack' is TRUE, it returns a raster stack containing 34 LCZ-related parameters. If 'istack' is FALSE, it returns a list of individual parameter rasters. If 'ishp' is TRUE, it returns an ESRI shapefile containing all parameters. If 'iselect' is specified, it returns the selected parameter(s) as a raster(s).

Author(s)

Max Anjos (<https://github.com/ByMaxAnjos>)

References

Stewart, I., and T. Oke, 2012: Local climate zones for urban temperature studies. Bull. Amer. Meteor. Soc., 93, 1879–1900. <https://doi.org/10.1175/BAMS-D-11-00019.1>

Examples

```
## Not run:
# Load the LCZ map
lcz_map <- raster::raster("path/to/lcz_map.tif")

# Get LCZ parameters as a raster stack
lcz_params_stack <- lcz_get_parameters(lcz_map, istack = TRUE)

# Get LCZ parameters as a list of individual parameter rasters
lcz_params_list <- lcz_get_parameters(lcz_map, istack = FALSE)

# Access individual parameters from the list
specific_param <- lcz_params_list$parameter_name

# Get specific parameters as a raster or shapefile
selected_params <- lcz_get_parameters(lcz_map, iselect = c("SVFmin"))

## End(Not run)
```

lcz_interp_eval

Evaluate the LCZ-Based Interpolated Map

Description

This function evaluates the variability of a spatial and temporal interpolation of a variable (e.g., air temperature) using LCZ as background. It supports both LCZ-based and conventional interpolation methods. The function allows for flexible time period selection, cross-validation, and station splitting for training and testing.

Usage

```
lcz_interp_eval(
  x,
  data_frame = "",
  var = "",
  station_id = "",
  ...,
  extract.method = "simple",
  LOOCV = TRUE,
  split.ratio = 0.8,
  sp.res = 100,
  tp.res = "hour",
  vg.model = "Sph",
  by = NULL,
  Anomaly = FALSE,
  impute = NULL,
  isave = FALSE,
  LCZinterp = TRUE
)
```

Arguments

<code>x</code>	A <code>SpatRaster</code> object containing the LCZ map. The LCZ map can be obtained using the <code>lcz_get_map()</code> functions.
<code>data_frame</code>	A data frame containing air temperature measurements and station IDs. The data frame must include a date field in hourly or higher resolution format.
<code>var</code>	A character string specifying the name of the variable to interpolate (e.g., "airT" for air temperature).
<code>station_id</code>	A character string specifying the name of the station ID variable in the data frame.
<code>...</code>	Additional arguments for the <code>selectByDate</code> function from the <code>openair</code> package. These arguments allow for flexible selection of specific time periods (year, month, day, hour). Examples include: <ul style="list-style-type: none"> • Year(s): Numeric value(s) specifying the year(s) to select. For example, <code>year = 1998:2004</code> selects all years between 1998 and 2004 (inclusive), while <code>year = c(1998, 2004)</code> selects only the years 1998 and 2004. • Month(s): Numeric or character value(s) specifying the months to select. Numeric examples: <code>month = 1:6</code> (January to June), or character examples: <code>month = c("January", "December")</code>. • Day(s): Numeric value(s) specifying the days to select. For instance, <code>day = 1:30</code> selects days from 1 to 30, or <code>day = 15</code> selects only the 15th day of the month. • Hour(s): Numeric value(s) specifying the hours to select. For example, <code>hour = 0:23</code> selects all hours in a day, while <code>hour = 9</code> selects only the 9th hour. • Start date: A string specifying the start date in either <code>start="DD/MM/YYYY"</code> (e.g., "1/2/1999") or "YYYY-mm-dd" format (e.g., "1999-02-01"). • End date: A string specifying the end date in either <code>end="DD/MM/YYYY"</code> (e.g., "1/2/1999") or "YYYY-mm-dd" format (e.g., "1999-02-01").
<code>extract.method</code>	A character string specifying the method used to assign the LCZ class to each station point. The default is "simple". Available methods are: <ul style="list-style-type: none"> • simple: Assigns the LCZ class based on the value of the raster cell in which the point falls. It often is used in low-density observational network. • two.step: Assigns LCZs to stations while filtering out those located in heterogeneous LCZ areas. This method requires that at least 80% of the pixels within a 5×5 kernel match the LCZ of the center pixel (Daniel et al., 2017). Note that this method reduces the number of stations. It often is used in ultra and high-density observational network, especially in LCZ classes with multiple stations. • bilinear: Interpolates the LCZ class values from the four nearest raster cells surrounding the point.
<code>LOOCV</code>	A logical value. If <code>TRUE</code> (default), leave-one-out cross-validation (LOOCV) is used for kriging. If <code>FALSE</code> , the split method into training and testing stations is used.

split.ratio	A numeric value representing the proportion of meteorological stations to be used for training (interpolation). The remaining stations will be used for testing (evaluation). For example, the default 0.8 indicates that 80% of the stations will be used for training and 20% for testing.
sp.res	A numeric value specifying the spatial resolution in meters for interpolation. The default is 100.
tp.res	A character string specifying the temporal resolution for averaging. The default is "hour". Other options include "day", "week", "month", "year", "season", "seasonyear", "monthyear", "weekday", "weekend", or custom intervals like "2 day", "2 week", "3 month", etc.
vg.model	A character string specifying the variogram model for kriging. The default is "Sph". Available models are: <ul style="list-style-type: none"> • Sph: Spherical model. • Exp: Exponential model. • Gau: Gaussian model. • Ste: M. Stein's parameterization.
by	data frame time-serie split: "year", "season", "seasonyear", "month", "monthyear", "weekday", "weekend", "site", "daylight", "dst" (daylight saving time). Note that the Daylight option may result in both daytime and nighttime hours being represented in UTC . See NOAA https://gml.noaa.gov/grad/solcalc/ and argument <i>type</i> in openair package: https://openair-project.github.io/book/sections/intro/openair-package.html#the-type-option
Anomaly	If TRUE, the anomalies are calculated. If FALSE (default) the raw air temperatures are used.
impute	A character string specifying the method to impute missing values. Options include "mean", "median", "knn", or "bag".
isave	A logical value. If TRUE, the plot is saved to the working directory.
LCZinterp	A logical value. If TRUE (default), the LCZ interpolation approach is used. If FALSE, conventional interpolation without LCZ is used.

Value

A summary table in CSV format containing time series of observed values, predicted values, and residuals. It also returns an ESRI shapefile (.gpkg) containing metadata of the interpolation.

Author(s)

Max Anjos (<https://github.com/ByMaxAnjos>)

References

Anjos, M., Targino, A. C., Krecl, P., Oukawa, G. Y. & Braga, R. F. Analysis of the urban heat island under different synoptic patterns using local climate zones. Build. Environ. 185, 107268 (2020). Fenner, D., Meier, F., Bechtel, B., Otto, M. & Scherer, D. Intra and inter 'local climate zone' variability of air temperature as observed by crowdsourced citizen weather stations in Berlin, Germany. Meteorol. Z. 26, 525–547 (2017). <http://www.gstat.org/>

Examples

```
## Not run:
# Evaluate air temperature interpolation values using Berlin data and LOOCV
data("lcz_data")
lcz_map <- lcz_get_map_generator(ID = "8576bde60bfe774e335190f2e8fdd125dd9f4299")
lcz_plot_map(lcz_map)
my_interp <- lcz_interp_eval(
  lcz_map, data_frame = lcz_data, var = "airT",
  station_id = "station", year = 2019, month = 9, day = 5,
  sp.res = 100, tp.res = "hour", LOOCV = TRUE,
  vg.model = "Sph", LCZinterp = TRUE, isave = TRUE
)

# Evaluate LCZ-based interpolation using split station (80% training, 20% testing)
my_interp <- lcz_interp_eval(
  lcz_map, data_frame = lcz_data, var = "airT",
  station_id = "station", tp.res = "hour", sp.res = 100,
  year = 2019, month = 9, day = 5, LOOCV = FALSE,
  split.ratio = 0.8, vg.model = "Sph", LCZinterp = TRUE, isave = TRUE
)

## End(Not run)
```

lcz_interp_map	<i>Interpolate spatial and temporal data using a Local Climate Zones (LCZ)</i>
----------------	--

Description

Generates an interpolated spatial map using LCZ as background. This function applies geostatistical interpolation techniques (e.g., kriging) to estimate variable values (e.g., air temperature) across the study area, integrating LCZ classifications. The function allows for flexible time period selection, like hour, daytime, nighttime and so on.

Usage

```
lcz_interp_map(
  x,
  data_frame = "",
  var = "",
  station_id = "",
  ...,
  extract.method = "simple",
  sp.res = 100,
  tp.res = "hour",
  vg.model = "Sph",
  by = NULL,
```

```

    impute = NULL,
    isave = FALSE,
    LCZinterp = TRUE
)

```

Arguments

x	A SpatRaster object representing the LCZ map, obtained via lcz_get_map() functions.
data_frame	A data frame containing air temperature measurements with station metadata. The data must include a date column in hourly or higher resolution.
var	Character. The variable to interpolate (e.g., "airT" for air temperature).
station_id	Character. The column name in data_frame identifying weather stations.
...	Additional arguments passed to selectByDate from the openair package for flexible time selection. Options include: <ul style="list-style-type: none"> • Year(s): year = 1998:2004 (selects 1998 to 2004) or year = c(1998, 2004). • Month(s): month = 1:6 (January to June) or month = c("January", "December"). • Day(s): day = 1:30 (days 1 to 30) or day = 15. • Hour(s): hour = 0:23 (all hours) or hour = 9. • Start/End Date: "YYYY-mm-dd" format (e.g., "1999-02-01").
extract.method	Character. The method used to assign LCZ classes to stations: <ul style="list-style-type: none"> • "simple": Assigns the LCZ based on the raster cell where the station is located. It often is used in low-density observational network. • "two.step": Filters out stations in heterogeneous LCZ areas (Daniel et al., 2017). It often is used in ultra and high-density observational network, especially in LCZ classes with multiple stations. • "bilinear": Uses bilinear interpolation to assign LCZ values.
sp.res	Numeric. Spatial resolution in meters (default: 100).
tp.res	Character. Temporal resolution for averaging ("hour", "day", etc.).
vg.model	Character. Variogram model for kriging. Default is "Sph" (spherical). Available models: <ul style="list-style-type: none"> • "Sph": Spherical model. • "Exp": Exponential model. • "Gau": Gaussian model. • "Ste": M. Stein's parameterization.
by	data frame time-series split: "year", "season", "seasonyear", "month", "monthyear", "weekday", "weekend", "site", "daylight", "dst" (daylight saving time). Note that the Daylight option may result in both daytime and nighttime hours being represented in UTC . See NOAA https://gml.noaa.gov/grad/solcalc/ and argument <i>type</i> in openair package: https://openair-project.github.io/book/sections/intro/openair-package.html#the-type-option
impute	Character. Method to impute missing values ("mean", "median", "knn", "bag").
isave	Logical. If TRUE, saves the plot to the working directory.
LCZinterp	Logical. If TRUE (default), applies LCZ-based interpolation.

Value

A spatially interpolated map in terra raster GeoTIFF format.

Author(s)

Max Anjos (<https://github.com/ByMaxAnjos>)

References

Anjos, M., Targino, A. C., Krecl, P., Oukawa, G. Y. & Braga, R. F. Analysis of the urban heat island under different synoptic patterns using local climate zones. *Build. Environ.* 185, 107268 (2020).
 Fenner, D., Meier, F., Bechtel, B., Otto, M. & Scherer, D. (2017). Intra and inter ‘local climate zone’ variability of air temperature as observed by crowdsourced citizen weather stations in Berlin, Germany. *Meteorol. Z.*, 26, 525–547. <http://www.gstat.org/>

See Also

See `lcz_inter_eval()` for LCZ interpolation evaluation.

Examples

```
## Not run:
# Load sample data and LCZ map
data("lcz_data")
lcz_map <- lcz_get_map_generator(ID = "8576bde60bfe774e335190f2e8fdd125dd9f4299")

# Plot the LCZ map
lcz_plot_map(lcz_map)

# Perform interpolation for 2020-01-02 at 04:00h
my_interp <- lcz_interp_map(
  lcz_map, data_frame = lcz_data, var = "airT",
  station_id = "station", year = 2020, month = 1, day = 2, hour = 4,
  sp.res = 100, tp.res = "hour", vg.model = "Sph",
  LCZinterp = TRUE
)

# Perform interpolation for an entire day (returns a raster stack of 24 hours)
my_interp <- lcz_interp_map(
  lcz_map, data_frame = lcz_data, var = "airT",
  station_id = "station", year = 2020, month = 1, day = 2,
  sp.res = 100, tp.res = "hour", vg.model = "Sph",
  LCZinterp = TRUE
)

## End(Not run)
```

lcz_plot_interp

*Visualize Interpolated LCZ map***Description**

This function plots the interpolated LCZ anomaly, LCZ air temperature, or other environmental variables.

Usage

```
lcz_plot_interp(
  x,
  palette = "muted",
  direction = 1,
  isave = FALSE,
  save_extension = "png",
  ...
)
```

Arguments

<code>x</code>	The LCZ map in SpatRaster format, either as a single raster or a raster stack, obtained from functions like <code>lcz_anomaly_map()</code> or <code>lcz_interp_map()</code> .
<code>palette</code>	Gradient palettes available in the tidyterra package. Default is "muted". More palettes can be found in the tidyterra documentation: https://dieghernan.github.io/tidyterra/articles/palettes/
<code>direction</code>	Sets order of colors. Default palette is 1. If direction is -1, palette color order is reversed
<code>isave</code>	Logical, indicating whether to save the plot to your directory. Default is FALSE.
<code>save_extension</code>	File format for saving the plot. Options: "png", "jpg", "jpeg", "tif", "pdf", "svg" (default is "png").
<code>...</code>	Additional arguments to modify axis, legend, and plot labels, including title, subtitle, and caption.

Value

A plot of the LCZ interpolated map in ggplot2 format.

See Also

See the documentation for `lcz_anomaly_map()` to obtain LCZ anomaly rasters and `lcz_interp_map()` for air temperature rasters.

Examples

```
## Not run:
# Plot the interpolated map
lcz_plot_interp(my_interp_map)

## End(Not run)
```

lcz_plot_map

*Visualize the LCZ Map***Description**

This function generates a graphical representation of a Local Climate Zone (LCZ) map provided as a SpatRaster object.

Usage

```
lcz_plot_map(
  x,
  isave = FALSE,
  show_legend = TRUE,
  save_extension = "png",
  inclusive = FALSE,
  ...
)
```

Arguments

x	A SpatRaster object containing the LCZ map to be plotted.
isave	Logical. Set to TRUE to save the plot in your working directory.
show_legend	Logical. If TRUE, displays the legend on the plot. If FALSE, the legend will be hidden. Default is TRUE.
save_extension	File format for saving the plot. Options: "png", "jpg", "jpeg", "tif", "pdf", "svg" (default is "png").
inclusive	Logical. Set to TRUE to use a colorblind-friendly palette.
...	Optional arguments to modify axis labels, legend, plot title, subtitle, and caption.

Value

A visual representation of the LCZ map in ggplot format.

See Also

See the documentation for `lcz_get_map()` to obtain an LCZ map.

Examples

```
## Not run:
lcz_plot_map(LCZmap)

## End(Not run)
```

lcz_plot_parameters	<i>Visualize LCZ parameters</i>
---------------------	---------------------------------

Description

This function plots the parameters of an Local Climate Zone map.

Usage

```
lcz_plot_parameters(
  x,
  iselect = "",
  all = FALSE,
  inclusive = FALSE,
  isave = FALSE,
  save_extension = "png",
  ...
)
```

Arguments

x	The LCZ map in SpatRaster in a stack format.
iselect	Code vector. Specify one or more parameter: <ul style="list-style-type: none"> • SVFmean: Mean Sky View Factor • SVFmax: Maximum Sky View Factor • SVFmin: Minimum Sky View Factor • ARmean: Mean Aspect Ratio • ARmax: Maximum Aspect Ratio • ARmin: Minimum Aspect Ratio • BSFmean: Mean Building Surface Fraction • BSFmax: Maximum Building Surface Fraction • BSFmin: Minimum Building Surface Fraction • ISFmean: Mean Impervious Surface Fraction • ISFmax: Maximum Impervious Surface Fraction • ISFmin: Minimum Impervious Surface Fraction • PSFmean: Mean Vegetation Surface Fraction • PSFmax: Maximum Vegetation Surface Fraction

	<ul style="list-style-type: none"> • PSFmin: Minimum Vegetation Surface Fraction • TSFmean: Mean Tree Surface Fraction • TSFmax: Maximum Tree Surface Fraction • TSFmin: Minimum Tree Surface Fraction • HREmean: Mean Height Roughness Elements • HREmax: Maximum Height Roughness Elements • HREmin: Minimum Height Roughness Elements • TRCmean: Mean Terrain Roughness class • TRCmax: Maximum Terrain Roughness class • TRCmin: Minimum Terrain Roughness class • SADmean: Mean Surface Admittance • SADmax: Maximum Surface Admittance • SADmin: Minimum Surface Admittance • SALmean: Mean Surface Albedo • SALmax: Maximum Surface Albedo • SALmin: Minimum Surface Albedo • AHmean: Mean Anthropogenic Heat Output • AHmax: Maximum Anthropogenic Heat Output • AHmin: Minimum Anthropogenic Heat Output • z0: Roughness Length class
all	Logical, specifying whether to save all selected parameters into LCZ4r_output. Default is FALSE.
inclusive	Set to TRUE to a colorblind-friendly palette.
isave	Logical, indicating whether to save the plot to your directory. Default is FALSE.
save_extension	File format for saving the plot. Options: "png", "jpg", "jpeg", "tif", "pdf", "svg" (default is "png").
...	An optional modify axis, legend, and plot labels: subtitle, and caption. It does not work with title.

Value

A plot of the selected LCZ parameters in ggplot2 format

#' @references Stewart, I., and T. Oke, 2012: Local climate zones for urban temperature studies. Bull. Amer. Meteor. Soc., 93, 1879–1900. <https://doi.org/10.1175/BAMS-D-11-00019.1>

See Also

See the documentation for `lcz_get_parameters()` to obtain LCZ parameters.

Examples

```
## Not run:
# Plot the mean Sky View Factor (SVFmean) for your city
lcz_plot_parameters(lcz_map, iselect = "SVFmean")
```

```
# Plot multiple parameters and save them to the LCZ4r_output directory
lcz_plot_parameters(lcz_map, iselect = c("BSFmean", "AHmean"))

## End(Not run)
```

lcz_ts

Analyze LCZ Time Series

Description

This function generates a graphical representation of time series air temperature data across different Local Climate Zones (LCZs). It allows for flexible temporal aggregation, faceting, smoothing, and visualization options including heatmaps, facet_plot line plots, and warming stripes. For more details on LCZ analysis, visit https://bymaxanjos.github.io/LCZ4r/articles/Introd_local_LCZ4r.html.

Usage

```
lcz_ts(
  x,
  data_frame = "",
  var = "",
  station_id = "",
  ...,
  time.freq = "hour",
  extract.method = "simple",
  plot_type = "basic_line",
  facet_plot = "LCZ",
  by = NULL,
  smooth = FALSE,
  impute = NULL,
  iplot = TRUE,
  isave = FALSE,
  save_extension = "png",
  palette = "VanGogh2",
  ylab = "Air temperature [Degree Celsius]",
  xlab = "Time",
  title = "LCZ - Timeseries",
  caption = "LCZ4r, 2024.",
  legend_name = "Temperature (C)"
)
```

Arguments

x	A SpatRaster object containing the LCZ map. The LCZ map can be obtained using the <code>lcz_get_map()</code> , <code>lcz_get_map_euro()</code> , and <code>lcz_get_map_usa()</code> functions.
---	--

<code>data_frame</code>	A data frame containing air temperature measurements and station IDs. The data frame should have a date field in hourly or higher resolution format.
<code>var</code>	The name of the variable in the data frame representing air temperature.
<code>station_id</code>	The name of the variable in the data frame representing station IDs.
<code>...</code>	Additional arguments for the <code>selectByDate</code> function from the <code>openair</code> package. These arguments allow for flexible selection of specific time periods (year, month, day, hour). Examples of how to use these arguments include: <ul style="list-style-type: none"> • Year(s): Numeric value(s) specifying the year(s) to select. For example, <code>year = 1998:2004</code> selects all years between 1998 and 2004 (inclusive), while <code>year = c(1998, 2004)</code> selects only the years 1998 and 2004. • Month(s): Numeric or character value(s) specifying the months to select. Numeric examples: <code>month = 1:6</code> (January to June), or character examples: <code>month = c("January", "December")</code>. • Day(s): Numeric value(s) specifying the days to select. For instance, <code>day = 1:30</code> selects days from 1 to 30, or <code>day = 15</code> selects only the 15th day of the month. • Hour(s): Numeric value(s) specifying the hours to select. For example, <code>hour = 0:23</code> selects all hours in a day, while <code>hour = 9</code> selects only the 9th hour. • Start date: A string specifying the start date in either <code>start="DD/MM/YYYY"</code> (e.g., "1/2/1999") or <code>"YYYY-mm-dd"</code> format (e.g., "1999-02-01"). • End date: A string specifying the end date in either <code>end="DD/MM/YYYY"</code> (e.g., "1/2/1999") or <code>"YYYY-mm-dd"</code> format (e.g., "1999-02-01").
<code>time.freq</code>	Defines the time period to average to. Default is "hour", but includes "day", "week", "month" or "year".
<code>extract.method</code>	A character string specifying the method used to assign the LCZ class to each station point. The default is "simple". Available methods are: <ul style="list-style-type: none"> • simple: Assigns the LCZ class based on the value of the raster cell in which the point falls. It often is used in low-density observational network. • two.step: Assigns LCZs to stations while filtering out those located in heterogeneous LCZ areas. This method requires that at least 80% of the pixels within a 5×5 kernel match the LCZ of the center pixel (Daniel et al., 2017). Note that this method reduces the number of stations. It often is used in ultra and high-density observational network, especially in LCZ classes with multiple stations. • bilinear: Interpolates the LCZ class values from the four nearest raster cells surrounding the point.
<code>plot_type</code>	A character string indicating the type of plot to generate. Options include: "basic_line", "facet_line", "heatmap", "warming_stripes". The default is "basic_line". Note: This argument does not work if the <code>by</code> argument is active.
<code>facet_plot</code>	A character string indicating whether to divide the plot by "LCZ" or "station". The default is "LCZ".
<code>by</code>	data frame time-serie split: "year", "season", "seasonyear", "month", "monthyear", "weekday", "weekend", "site", "daylight", "dst" (daylight saving time).

Note that the **Daylight option may result in both daytime and nighttime hours being represented in UTC**. See NOAA <https://gml.noaa.gov/grad/solcalc/> and argument *type* in openair package: <https://openair-project.github.io/book/sections/intro/openair-package.html#the-type-option>

smooth	Logical. Set to TRUE to add a smoothed trend line using a generalized additive model (GAM). The default is FALSE.
impute	Method to impute missing values ("mean", "median", "knn", "bag").
iplot	Set to TRUE to return a plot. If FALSE, a data frame is returned.
isave	Set to TRUE to save all results (plot, time-series) into your directory.
save_extension	A character string indicating the file format for saving the plot. Options include: "png", "jpg", "jpeg", "tif", "pdf", "svg". The default is "png".
palette	A character string specifying the color palette to use. The default is "Van-Gogh2". You can choose from palettes available in MetBrewer: https://github.com/BlakeRMills/MetBrewer?tab=readme-ov-file#palettes .
ylab	A character string for the y-axis label. The default is the variable name.
xlab	A character string for the x-axis label. The default is "Time".
title	A character string for the plot title. The default is "".
caption	source data. Default can be "Source:LCZ4r,2024".
legend_name	Legend name for heatmap and warming stripes plots. Default is "Temperature (°C)".

Value

A ggplot object representing the time series of air temperature in different LCZs, or a data frame in CSV format if `iplot = FALSE`.

Author(s)

Max Anjos (<https://github.com/ByMaxAnjos>)

See Also

See the documentation for `lcz_get_map()`, `lcz_get_map_euro()`, and `lcz_get_map_usa()` to obtain an LCZ map.

Examples

```
## Not run:
# Hourly air temperature values in 2019.
my_ts <- lcz_ts(lcz_map, df = lcz_data, var = "airT",
               station_id = "station", year = 2020)

## End(Not run)
```

lcz_tutorial	<i>LCZ4r Tutorials</i>
--------------	------------------------

Description

Opens LCZ4r tutorial in a web browser. This function requires an internet connection.

Usage

```
lcz_tutorial(x = "general")
```

Arguments

x	Character string. Either "general" for general functions or "local" for local functions. Default is "general".
---	--

Value

Opens the LCZ4r tutorial in a web browser.

Examples

```
## Not run:
lcz_tutorial("general")
lcz_tutorial("local")

## End(Not run)
```

lcz_uhi_intensity	<i>Assess the Urban Heat Island Intensity</i>
-------------------	---

Description

This function calculates the Urban Heat Island (UHI) intensity based on air temperature measurements and Local Climate Zones (LCZ). More details: https://bymaxanjos.github.io/LCZ4r/articles/Introd_local_LCZ4r.html

Usage

```
lcz_uhi_intensity(
  x,
  data_frame = "",
  var = "",
  station_id = "",
  ...,
  time.freq = "hour",
```

```

extract.method = "simple",
method = "LCZ",
Turban = NULL,
Trural = NULL,
by = NULL,
impute = NULL,
group = FALSE,
iplot = TRUE,
isave = FALSE,
save_extension = "png",
ylab = "Air temperature [C]",
xlab = "Time",
ylab2 = "UHI",
title = "",
caption = "LCZ4r,2024"
)

```

Arguments

<code>x</code>	A <code>SpatRaster</code> object containing the LCZ map. The LCZ map can be obtained using the <code>lcz_get_map()</code> function.
<code>data_frame</code>	A data frame containing air temperature measurements and station IDs. The data frame should have a date field in hourly or higher resolution format.
<code>var</code>	The name of the variable in the data frame representing air temperature.
<code>station_id</code>	The name of the variable in the data frame representing station IDs.
<code>...</code>	Additional arguments for the <code>selectByDate</code> function from the <code>openair</code> package. These arguments allow for flexible selection of specific time periods (year, month, day, hour). Examples of how to use these arguments include: <ul style="list-style-type: none"> • Year(s): Numeric value(s) specifying the year(s) to select. For example, <code>year = 1998:2004</code> selects all years between 1998 and 2004 (inclusive), while <code>year = c(1998, 2004)</code> selects only the years 1998 and 2004. • Month(s): Numeric or character value(s) specifying the months to select. Numeric examples: <code>month = 1:6</code> (January to June), or character examples: <code>month = c("January", "December")</code>. • Day(s): Numeric value(s) specifying the days to select. For instance, <code>day = 1:30</code> selects days from 1 to 30, or <code>day = 15</code> selects only the 15th day of the month. • Hour(s): Numeric value(s) specifying the hours to select. For example, <code>hour = 0:23</code> selects all hours in a day, while <code>hour = 9</code> selects only the 9th hour. • Start date: A string specifying the start date in either <code>start="DD/MM/YYYY"</code> (e.g., "1/2/1999") or <code>"YYYY-mm-dd"</code> format (e.g., "1999-02-01"). • End date: A string specifying the start date in either <code>end="DD/MM/YYYY"</code> (e.g., "1/2/1999") or <code>"YYYY-mm-dd"</code> format (e.g., "1999-02-01").
<code>time.freq</code>	The time period to average to. Default is "hour", but can also be "day", "week", "month", or "year".

extract.method	<p>A character string specifying the method used to assign the LCZ class to each station point. The default is "simple". Available methods are:</p> <ul style="list-style-type: none"> • simple: Assigns the LCZ class based on the value of the raster cell in which the point falls. It often is used in low-density observational network. • two.step: Assigns LCZs to stations while filtering out those located in heterogeneous LCZ areas. This method requires that at least 80% of the pixels within a 5×5 kernel match the LCZ of the center pixel (Daniel et al., 2017). Note that this method reduces the number of stations. It often is used in ultra and high-density observational network, especially in LCZ classes with multiple stations. • bilinear: Interpolates the LCZ class values from the four nearest raster cells surrounding the point.
method	Method to calculate the UHI intensity. Options include: LCZ and manual. The LCZ method automatically identifies the LCZ build types, starting from LCZ 1 and progressing to LCZ 10, to represent the urban temperature, then it starts from LCZ natural LCZ (11-16) to represent the rural temperature. The manual method is a character string indicating the stations as references for the urban and rural areas.
Turban	Urban station references, if the method "manual" is selected.
Trural	Rural station references, if the method "manual" is selected.
by	data frame time-series split: "year", "season", "seasonyear", "month", "monthyear", "weekday", "weekend", "site", "daylight", "dst" (daylight saving time). Note that the Daylight option may result in both daytime and nighttime hours being represented in UTC . See NOAA https://gml.noaa.gov/grad/solcalc/ and argument <i>type</i> in openair package: https://openair-project.github.io/book/sections/intro/openair-package.html#the-type-option
impute	Method to impute missing values. Options include "mean", "median", "knn", or "bag".
group	If TRUE urban, rural and UHI - related temperatures are grouped in the same plot.
iplot	Set to TRUE to return a plot. If FALSE, a data frame is returned.
isave	Set to TRUE to save all results (plot, time-series, station references) into your directory.
save_extension	A character string indicating the file format for saving the plot. Options include: "png", "jpg", "jpeg", "tif", "pdf", "svg". The default is "png".
ylab	The label for the y-axis.
xlab	The label for the x-axis. Default is "Time".
ylab2	The label for the right y-axis.
title	The title of the plot. Default is an empty string.
caption	The source data caption.

Value

A visual representation of the time series of UHI in ggplot or data frame .csv format.

Author(s)

Max Anjos (<https://github.com/ByMaxAnjos>)

See Also

See the documentation for `lcz_get_map()` to obtain an LCZ map.

Examples

```
## Not run:  
# Calculate hourly UHI intensity for 2019 using the LCZ method  
my_uhi <- lcz_uhi_intensity(lcz_map, data_frame = lcz_data, var = "airT",  
                           station_id = "station", year = 2019,  
                           time.freq = "hour", method = "LCZ")  
  
## End(Not run)
```

Index

* Climate

- lcz_anomaly, [2](#)
- lcz_anomaly_map, [5](#)
- lcz_get_map, [10](#)
- lcz_get_map2, [14](#)
- lcz_get_map_euro, [11](#)
- lcz_get_map_generator, [12](#)
- lcz_get_map_usa, [13](#)
- lcz_get_parameters, [15](#)
- lcz_interp_eval, [17](#)
- lcz_interp_map, [20](#)
- lcz_plot_interp, [23](#)
- lcz_plot_map, [24](#)
- lcz_plot_parameters, [25](#)
- lcz_ts, [27](#)
- lcz_uhi_intensity, [30](#)

* Generator

- lcz_get_map_generator, [12](#)

* LCZ,

- lcz_anomaly, [2](#)
- lcz_get_map, [10](#)
- lcz_get_map2, [14](#)
- lcz_get_map_euro, [11](#)
- lcz_get_map_generator, [12](#)
- lcz_get_map_usa, [13](#)
- lcz_get_parameters, [15](#)
- lcz_plot_interp, [23](#)
- lcz_plot_map, [24](#)
- lcz_plot_parameters, [25](#)
- lcz_ts, [27](#)
- lcz_uhi_intensity, [30](#)

* LCZ

- lcz_get_map_generator, [12](#)

* Local

- lcz_anomaly, [2](#)
- lcz_anomaly_map, [5](#)
- lcz_get_map, [10](#)
- lcz_get_map2, [14](#)
- lcz_get_map_euro, [11](#)

- lcz_get_map_generator, [12](#)
- lcz_get_map_usa, [13](#)
- lcz_get_parameters, [15](#)
- lcz_interp_eval, [17](#)
- lcz_interp_map, [20](#)
- lcz_plot_interp, [23](#)
- lcz_plot_map, [24](#)
- lcz_plot_parameters, [25](#)
- lcz_ts, [27](#)
- lcz_uhi_intensity, [30](#)

* Zone,

- lcz_anomaly, [2](#)
- lcz_anomaly_map, [5](#)
- lcz_get_map, [10](#)
- lcz_get_map2, [14](#)
- lcz_get_map_euro, [11](#)
- lcz_get_map_generator, [12](#)
- lcz_get_map_usa, [13](#)
- lcz_get_parameters, [15](#)
- lcz_interp_eval, [17](#)
- lcz_interp_map, [20](#)
- lcz_plot_interp, [23](#)
- lcz_plot_map, [24](#)
- lcz_plot_parameters, [25](#)
- lcz_ts, [27](#)
- lcz_uhi_intensity, [30](#)

* analysis

- lcz_anomaly, [2](#)
- lcz_get_map, [10](#)
- lcz_get_map2, [14](#)
- lcz_get_map_euro, [11](#)
- lcz_get_map_usa, [13](#)
- lcz_get_parameters, [15](#)
- lcz_plot_interp, [23](#)
- lcz_plot_map, [24](#)
- lcz_plot_parameters, [25](#)
- lcz_ts, [27](#)
- lcz_uhi_intensity, [30](#)

* anomaly,

- lcz_anomaly_map, 5
- * **climate**,
 - lcz_anomaly, 2
 - lcz_get_map, 10
 - lcz_get_map2, 14
 - lcz_get_map_euro, 11
 - lcz_get_map_generator, 12
 - lcz_get_map_usa, 13
 - lcz_get_parameters, 15
 - lcz_plot_interp, 23
 - lcz_plot_map, 24
 - lcz_plot_parameters, 25
 - lcz_ts, 27
 - lcz_uhi_intensity, 30
- * **datasets**
 - lcz_data, 9
- * **interpolation**,
 - lcz_anomaly_map, 5
 - lcz_interp_eval, 17
 - lcz_interp_map, 20
- * **kriging**.
 - lcz_anomaly_map, 5
 - lcz_interp_map, 20
- * **kriging**
 - lcz_interp_eval, 17
- * **spatial**
 - lcz_anomaly, 2
 - lcz_get_map, 10
 - lcz_get_map2, 14
 - lcz_get_map_euro, 11
 - lcz_get_map_usa, 13
 - lcz_get_parameters, 15
 - lcz_plot_interp, 23
 - lcz_plot_map, 24
 - lcz_plot_parameters, 25
 - lcz_ts, 27
 - lcz_uhi_intensity, 30
- * **urban**
 - lcz_anomaly, 2
 - lcz_get_map, 10
 - lcz_get_map2, 14
 - lcz_get_map_euro, 11
 - lcz_get_map_generator, 12
 - lcz_get_map_usa, 13
 - lcz_get_parameters, 15
 - lcz_plot_interp, 23
 - lcz_plot_map, 24
 - lcz_plot_parameters, 25
- lcz_ts, 27
- lcz_uhi_intensity, 30
- lcz_anomaly, 2
- lcz_anomaly_map, 5
- lcz_cal_area, 8
- lcz_data, 9
- lcz_get_map, 10
- lcz_get_map2, 14
- lcz_get_map_euro, 11
- lcz_get_map_generator, 12
- lcz_get_map_usa, 13
- lcz_get_parameters, 15
- lcz_interp_eval, 17
- lcz_interp_map, 20
- lcz_plot_interp, 23
- lcz_plot_map, 24
- lcz_plot_parameters, 25
- lcz_ts, 27
- lcz_tutorial, 30
- lcz_uhi_intensity, 30