

# Package ‘climenv’

November 2, 2023

**Type** Package

**Title** Download, Extract and Visualise Climate and Elevation Data

**Version** 1.0.0

**Description** Grants access to three widely recognised modelled data sets, namely Global Climate Data (WorldClim 2), Climatologies at high resolution for the earth's land surface areas (CHELSA), and National Aeronautics and Space Administration's (NASA) Shuttle Radar Topography Mission (SRTM). It handles both multi and single geospatial polygon and point data, extracts outputs that can serve as covariates in various ecological studies. Provides two common graphic options – the Walter-Lieth (1960) [https://donum.uliege.be/bitstream/2268.1/7079/1/Walter-Lieth\\_Klimadiagramm-Weltatlas.pdf](https://donum.uliege.be/bitstream/2268.1/7079/1/Walter-Lieth_Klimadiagramm-Weltatlas.pdf) climate diagram and the Holdridge (1967) [https://reddcr.go.cr/sites/default/files/centro-de-documentacion/holdridge\\_1966\\_-\\_life\\_zone\\_ecology.pdf](https://reddcr.go.cr/sites/default/files/centro-de-documentacion/holdridge_1966_-_life_zone_ecology.pdf) life zone classification scheme. Provides one new graphic scheme of our own design which incorporates aspects of both Walter-Leigh and Holdridge. Provides user-friendly access and extraction of globally recognisable data sets to enhance their usability across a broad spectrum of applications.

**License** GPL-3

**BugReports** <https://github.com/jamestsakalos/climenv/issues>

**URL** <https://github.com/jamestsakalos/climenv>

**Encoding** UTF-8

**Language** en-GB

**LazyData** true

**VignetteBuilder** knitr

**Depends** R (>= 4.1.0)

**Imports** climaemet, dismo, dplyr, elevatr (>= 0.99), exactextractr, geodata, glue, methods, plyr, randomForest, sf, sp, Ternary, terra

**Suggests** covr, fs, knitr, progress, raster, rmarkdown, testthat (>= 0.2.0), vdiff

**Config/Needs/check** rcmdcheck

**Config/Needs/coverage** covr

**Config/Needs/website** pkgdown

**Config/testthat/edition** 3

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** James L. Tsakalos [aut, cre, cph]

(<https://orcid.org/0000-0001-5067-196X>),

Martin R. Smith [aut] (<https://orcid.org/0000-0001-5660-1727>),

Federico Luebert [aut] (<https://orcid.org/0000-0003-2251-4056>),

Ladislav Mucina [aut] (<https://orcid.org/0000-0003-0317-8886>)

**Maintainer** James L. Tsakalos <jamestsakalos@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-11-02 19:40:02 UTC

## R topics documented:

bioclimate . . . . .	2
ce_download . . . . .	3
ce_extract . . . . .	5
chelsa . . . . .	7
elev . . . . .	8
it_data . . . . .	10
it_data_extended . . . . .	11
it_pt . . . . .	11
it_py . . . . .	12
plot_c . . . . .	13
plot_h . . . . .	15
plot_wl . . . . .	16
srtm_tiles . . . . .	17
worldclim . . . . .	18
<b>Index</b>	<b>20</b>

---

bioclimate	<i>Holdridge points</i>
------------	-------------------------

---

### Description

Calculates the values of bioclimatic indices from monthly temperature and precipitation measurements, following Szelepcsényi et al. 2014.

### Usage

```
bioclimate(temp, prec)
```

**Arguments**

temp	A matrix or data frame with 12 columns containing temperatures, in degrees C, for each month of the calendar year.
prec	A matrix or data frame with the same dimensions as temp, containing precipitation measurements in mm for each month of the year.

**Value**

bioclimate() returns a data.frame whose columns correspond to:

- abt, mean annual biotemperature; mean annual temperature after replacing entries outside the range 0° to 30° with zeros;
- tap, total annual precipitation (mm);
- per, potential evapotranspiration ratio

**Author(s)**

Martin R. Smith

**References**

Szelepcsényi Z, Breuer H, Sümegei P (2014) The climate of Carpathian Region in the 20th century based on the original and modified Holdridge life zone system. *Cent Eur J Geosci* 6, 293–307. [doi:10.2478/s1353301201895](https://doi.org/10.2478/s1353301201895)

Holdridge L. R. (1959) Simple method for determining potential evapotranspiration from temperature data. *Science*, 130, 572. [doi:10.1126/science.130.3375.572](https://doi.org/10.1126/science.130.3375.572)

---

ce\_download

*Download climate and elevation data*

---

**Description**

ce\_download() brings together the worldclim(), chelsa() and elev() functions to streamline downloading.

**Usage**

```
ce_download(  
  output_dir,  
  c_source = "WorldClim",  
  e_source = "mapzen",  
  var = "all",  
  location,  
  ...  
)
```

**Arguments**

output_dir	Character (e.g., "../Desktop/chelsa"). Pathway to where the data will be stored.
c_source	Character (e.g., "CHELSA or WorldClim"). Indicating the climate data source.
e_source	Character (e.g., mapzen or geodata). Indicating the elevation data source.
var	Character. If supplied will download a subset of the climate data. Must be one of "all" (default), "prec", "tmax", "tmin" or "tmean" to download the corresponding climate data.
location	A "sp", "sf" polygon or point object. See <a href="#">sf::st_polygon</a> to make polygons and <a href="#">sf::st_as_sf</a> to make point objects.
...	Arguments to control a download from the Internet download.file().

**Value**

See documentation from [chelsa\(\)](#), [worldclim\(\)](#) and [elev\(\)](#).

**Author(s)**

James L. Tsakalos and Martin R. Smith

**See Also**

The underlying functions [chelsa\(\)](#), [worldclim\(\)](#) and [elev\(\)](#).

**Examples**

```
# Download time will depend on the size of the area you wish to access
# climate data for and your internet connection speed.

# Make a polygon file
regents <- sf::st_polygon(
  list(
    cbind(
      "lon" = c(51.537, 51.525, 51.523, 51.530, 51.534, 51.537),
      "lat" = c(-0.150, -0.145, -0.156, -0.167, -0.163, -0.150)
    )
  )
)

# Create temporary output directory
temp_path <- tempdir()

# Run the download function
ce_download(
  output_dir = temp_path,
  location = regents
)
```

```
# Reset user options
unlink(file.path(temp_path))
```

---

ce\_extract

*ce\_extract*


---

## Description

Extracts climate and/or elevation data for generated over an area or at fixed point/s.

## Usage

```
ce_extract(
  path = NULL,
  location = NULL,
  location_g = NULL,
  c_source = "WorldClim",
  var = "all"
)
```

## Arguments

path	Character. Path to folder containing the climate and elevation raster data. The folders should be named prec, tmax, tavg and tmin and elev. Within each climate folder there should be 12 raster files (e.g. "prec01.tif") corresponding to the monthly climate data. There should be only one raster file within the elev subfolder named "srtm.tif".
location	A "sp", "sf" polygon or point object. See <a href="#">sf::st_polygon</a> to make polygons and <a href="#">sf::st_as_sf</a> to make point objects.
location_g	Character. Informs how the zonal statistics are exported. Must correspond to a column of the "location" argument. If NULL, the zonal statistics are calculated for all features of "location" and a warning issued.
c_source	Character (e.g., "CHELSA or WorldClim"). Indicating the climate data source.
var	Character. If supplied will download a subset of the climate data. Must be one of "all" (default), "prec", "tmax", "tmin" or "tmean" to download the corresponding climate data.

## Value

Returns a list storing matrices containing the mean and standard deviation of the climate and/or elevation data. Each column represents a month, each row represents a feature of the location sp, sf polygon or point object. Values returned are either degrees Celsius for (tmax, tavg, tmin) or mm (prec).

**Author(s)**

James L. Tsakalos and Martin R. Smith

**See Also**

The downloading ([ce\\_download\(\)](#)), and the plotting ([plot\\_h\(\)](#) & [plot\\_wl\(\)](#)) functions.

**Examples**

```
# Create some random data

# Create temporary file
temp_path <- tempfile()
on.exit(unlink(file.path(temp_path)), add = TRUE)

# Create the required subdirectories
dir.create(file.path(temp_path, "/elev"), recursive = TRUE)
dir.create(file.path(temp_path, "/prec"), recursive = TRUE)
dir.create(file.path(temp_path, "/tmax"), recursive = TRUE)
dir.create(file.path(temp_path, "/tavg"), recursive = TRUE)
dir.create(file.path(temp_path, "/tmin"), recursive = TRUE)

# Create an empty raster
r <- terra::rast(ncol = 10, nrow = 10)

# Modify the base Raster
#* Elevation 100m ####
terra::values(r) <- 1:100
terra::writeRaster(r, paste0(temp_path, "/elev/srtm.tif"))

# create and save precipitation and temperature rasters ####
x <- c(5, 10, 15, 20, 25, 34, 25, 20, 15, 10, 5, 0) * 8
for (i in sprintf("%02d", 1:12)) {
  terra::writeRaster(r, paste0(temp_path, paste0("/prec/prec_", i, ".tif")))
  terra::writeRaster(r, paste0(temp_path, paste0("/tmax/tmax_", i, ".tif")))
  terra::writeRaster(r, paste0(temp_path, paste0("/tmin/tmin_", i, ".tif")))
  terra::writeRaster(r, paste0(temp_path, paste0("/tavg/tavg_", i, ".tif")))
}

# Create a polygon file from the raster
terra::values(r) <- 1:100
pol_py <- sf::st_as_sf(terra::as.polygons(r))
pol_py$grp <- c(rep("low", 25), rep("high", 75))

# Run the download function
ce_extract(
  path = temp_path,
  location = pol_py,
  location_g = "grp"
)
```

---

chelsa *Download CHELSA climate data*

---

## Description

chelsa() downloads the CHELSA (Climatologies at high resolution for the earth's land surface areas) V2.1 data of temperature and precipitation at a resolution of 30 arc sections.

## Usage

```
chelsa(output_dir = NULL, var = "all", quiet = FALSE, ...)
```

## Arguments

output_dir	Character (e.g., "../Desktop/chelsa"). Pathway to where the data will be stored.
var	Character. If supplied will download a subset of the climate data. Must be one of "all" (default), "prec", "tmax", "tmin" or "tmean" to download the corresponding climate data.
quiet, ...	Arguments to control a download from the Internet download.file().

## Value

Returns four subfolders named prec, tmax, tmin and tmean. Each folder contains 12 GeoTiff (.tif) files, one for each month of the year for the time period 1981–2010. Each of the files are downloaded at a spatial resolution of 30 arc seconds (~1 km sq.). The precipitation folder contains average monthly precipitation (mm). The tmax folder contains maximum monthly temperature. The tmin folder contains minimum monthly temperature. The tmean folder contains the average monthly temperature. The unit of measure for temperature is in °C.

## Author(s)

James L. Tsakalos and Martin R. Smith

## References

- Karger, D.N., Conrad, O., Böhner, J., Kawohl, T., Kreft, H., Soria-Auza, R.W. et al (2017) Climatologies at high resolution for the earth's land surface areas. *Scientific Data*, 4, 170122. doi:10.1038/sdata.2017.122
- Karger, D.N., Conrad, O., Böhner, J., Kawohl, T., Kreft, H., Soria-Auza, R.W. et al. (2021) Climatologies at high resolution for the earth's land surface areas. *EnviDat*. doi:10.16904/envidat.228.v2.1

## See Also

Downloading from WorldClim V2.1 [worldclim\(\)](#) or a more convenient function for other climate and elevation data [ce\\_download\(\)](#).

## Examples

```
# Download time will depend on the size of the area you wish to access
# climate data for and your internet connection speed.

# Create temporary file
temp_path <- tempfile()

# Download the WorldClim data
chelsa(
  output_dir = temp_path
)

# Reset user options
unlink(file.path(temp_path))

# Note that unlike worldclim() we do not specify the location argument
# because it is not yet possible to extract smaller tile sections
```

---

elev

*Download elevation data*

---

## Description

`elev()` downloads elevation data the Shuttle Radar Topography Mission (SRTM), specifically the hole-filled CGIAR-SRTM (90 m resolution) for latitudes between -60 and 60 or Mapzen's synthesis digital elevation product.

## Usage

```
elev(output_dir, location, e_source = "mapzen", verbose = FALSE, ...)
```

## Arguments

<code>output_dir</code>	Character (e.g., <code>"../Desktop/chelsa"</code> ). Pathway to where the data will be stored.
<code>location</code>	A <code>"sp"</code> , <code>"sf"</code> polygon or point object. See <a href="#">sf::st_polygon</a> to make polygons and <a href="#">sf::st_as_sf</a> to make point objects.
<code>e_source</code>	Character (e.g., <code>mapzen</code> or <code>geodata</code> ). Indicating the elevation data source.
<code>verbose</code>	Logical specifying whether to display verbose output when downloading from Mapzen.
<code>...</code>	Additional arguments to <a href="#">download.file()</a> .



**Value**

`elev()` is called for its side-effects. It invisibly returns a "SpatRaster" object if files were downloaded successfully, and returns NULL otherwise. If the elevation data is sourced from geodata the SpatRaster is downloaded at a spatial resolution of 30 arc seconds (~1 km sq.). If elevation data is from Mapzen then the SpatRaster will be a mosaic. Specifically, Mapzen's SpatRaster is unique as it combines several sources of digital elevation models, including SRTM, the ArcticDEM (covering all areas north of 60°), EUDEM (digital elevation model over Europe; for review, see Mouratidis & Ampatzidis, 2019), and others into a single product. The resolution of this SpatRaster was set to 7, corresponding to 611.5 m ground resolution at 60° latitude 864.8 m at 45° and 1223 m at 0°.

**Author(s)**

James L. Tsakalos and Martin R. Smith

**References**

Hijmans, R.J., Barbosa, M., Ghosh, A., & Mandel, A. (2023). geodata: Download Geographic Data. R package version 0.5-8. <https://CRAN.R-project.org/package=geodata>

Hollister, J. (2022). elevatr: Access Elevation Data from Various APIs. R package version 1.0.0. [doi:10.5281/zenodo.5809645](https://doi.org/10.5281/zenodo.5809645) <https://CRAN.R-project.org/package=elevatr>

Mouratidis, A., & Ampatzidis, D. (2019). European Digital Elevation Model Validation against Extensive Global Navigation Satellite Systems Data and Comparison with SRTM DEM and ASTER GDEM in Central Macedonia (Greece). *ISPRS International Journal of Geo-Information* 8, 108. [doi:10.3390/ijgi8030108](https://doi.org/10.3390/ijgi8030108)

**See Also**

A more convenient function for other climate and elevation data [ce\\_download\(\)](#). See [sf::st\\_polygon](#) to make polygons and [sf::st\\_as\\_sf](#) to make point objects.

**Examples**

```
# Make a small polygon
polygon_py_sm <- sf::st_polygon(
  list(cbind(long = c(156, 156, 155, 156),
    lat = c(-60, -59, -60, -60)))
)
polygon_py_sm <- sf::st_geometry(polygon_py_sm)
sf::st_crs(polygon_py_sm) <- "epsg:4326"

# Create temporary file
temp_path <- tempfile()

# elevation will be saved in the output_dir (i.e. output directory)
elev(output_dir = temp_path, location = polygon_py_sm)

# Reset user options
unlink(file.path(temp_path))
```

---

`it_data`*Italian Biome extracted data*

---

### Description

The package contains climatic data extracted for the Biomes of Italy.

### Usage

```
it_data
```

### Format

An object of class `list` of length 12.

### Details

Returns a 11 data frames and one 'Readme' note. Eight feature climate data (i.e., one each for the mean and standard deviation of `tmax`, `tmean`, `tmin` and `prec`). Within each data frame the columns represent a month (Jan–Dec), each row represents a feature of the location `sp`, `sf` polygon or point object. Values returned are either degrees Celsius for (`tmax`, `tmean`, `tmin`) or mm (`prec`). One data frame features elevation data. Within this data frame, one column shows the mean and the second the standard deviation. One data frame contains the central latitude of each feature. One, called "abmt" features the absolute minimum temperature for each month. For more details on the assignment of Biomes please refer to Mucina et al. (2023).

### References

Mucina, L., Divišek, J., & Tsakalos, J.L. (2023) Europe, Ecosystems of. In: Encyclopedia of biodiversity, vol X (in print). doi:[10.1016/B9780128225622.000591](https://doi.org/10.1016/B9780128225622.000591)

### Examples

```
data("it_data", package = "climenv")
head(it_data$abmt)
```

---

it_data_extended	<i>Italian Biome extracted data</i>
------------------	-------------------------------------

---

**Description**

The package contains climatic data extracted for the Biomes of Italy.

**Usage**

```
it_data_extended
```

**Format**

An object of class `list` of length 12.

**Details**

Returns a 11 data frames and one 'Readme' note. Eight feature climate data (i.e., one each for the mean and standard deviation of `tmax`, `tmean`, `tmin` and `prec`). Within each data frame the columns represent a month (Jan–Dec), each row represents a feature of the location `sp`, `sf` polygon or point object. Values returned are either degrees Celsius for (`tmax`, `tmean`, `tmin`) or mm (`prec`). One data frame features elevation data. Within this data frame, one column shows the mean and the second the standard deviation. One data frame contains the central latitude of each feature. One, called "abmt" features the absolute minimum temperature for each month. For more details on the assignment of Biomes please refer to Mucina et al. (2023).

**References**

Mucina, L., Divišek, J., & Tsakalos, J.L. (2023) Europe, Ecosystems of. In: Encyclopedia of biodiversity, vol X (in print). doi:[10.1016/B9780128225622.000591](https://doi.org/10.1016/B9780128225622.000591)

**Examples**

```
data("it_data_extended", package = "climenv")
head(it_data_extended$abmt)
```

---

it_pt	<i>Italian Biome point data</i>
-------	---------------------------------

---

**Description**

The package contains geospatial point data showing the Biomes of Italy.

**Usage**

```
it_pt
```

**Format**

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 9963 rows and 2 columns.

**Details**

The first column, `geometry`, refers to the X and Y coordinate of each point spaced over Italy. The second column, `GB`, refers to the Global Biome assigned to each coordinate. For more details on the assignment of Biomes please refer to Mucina et al. (2023).

**References**

Mucina, L., Divišek, J., & Tsakalos, J.L. (2023) Europe, Ecosystems of. In: Encyclopedia of biodiversity, vol X (in print). doi:[10.1016/B9780128225622.000591](https://doi.org/10.1016/B9780128225622.000591)

**Examples**

```
data("it_pt", package = "climenv")
plot(it_pt)
```

---

it\_py

*Italian Biome polygon data*

---

**Description**

The package contains geospatial polygon data showing the Biomes of Italy.

**Usage**

```
it_py
```

**Format**

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 328 rows and 2 columns.

**Details**

The first column, `geometry`, refers to the X and Y coordinate of each polygon spaced over Italy. The second column, `GB`, refers to the Global Biome assigned to each coordinate. For more details on the assignment of Biomes please refer to Mucina et al. (2023).

**References**

Mucina, L., Divišek, J., & Tsakalos, J.L. (2023) Europe, Ecosystems of. In: Encyclopedia of biodiversity, vol X (in print). doi:[10.1016/B9780128225622.000591](https://doi.org/10.1016/B9780128225622.000591)

**Examples**

```
data("it_py", package = "climenv")
plot(it_py)
```

---

 plot\_c
 

---



---

 plot\_c
 

---

### Description

Creates a graph using the climate and elevation data which has been extracted for a given location. It accepts the data formatted from the `ce_extract` function.

### Usage

```
plot_c(
  data,
  geo_id,
  interval_prec = 20,
  interval_temp = 5,
  stretch_temp = 4,
  p_cex = 1,
  nchr_main = 32,
  l_main = 0.1,
  l_prec = 0,
  l_month = -0.6,
  lwd_temp = 1.5,
  lwd_prec = 9,
  l_units = 0.1,
  l_tcols = c(14.5, 16.5, 18.5, 19.5, 22.5)
)
```

### Arguments

<code>data</code>	List. Containing climate, elevation and latitude data sets. Structured by <code>ce_extract()</code> .
<code>geo_id</code>	Character. Corresponding to a specific feature contained in the "location" argument (specifically in the "location_g" column).
<code>interval_prec</code>	Numeric. Interval of the precipitation axis.
<code>interval_temp</code>	Numeric. Interval of the temperature axis.
<code>stretch_temp</code>	Numeric. Ratio for stretching temperature relative to precipitation.
<code>p_cex</code>	Numeric. Text size for entire plot.
<code>nchr_main</code>	Numeric. Controls the number of characters of the main title before wrapping to a new line.
<code>l_main</code>	Numeric. Line position of the main title.
<code>l_prec</code>	Numeric. Line position of the y axis (precipitation).
<code>l_month</code>	Numeric. Line position of the x axis (months).
<code>lwd_temp</code>	Numeric. Line width of temperature.
<code>lwd_prec</code>	Numeric. Line width of precipitation.

`l_units` Numeric. Line position of precipitation and temperature units.  
`l_tcols` List. Line position of the table columns. Must be length 5 corresponding to the position (left to right) for each column.

**Value**

Returns a base R family of plot. This function uses the **dismo** package to calculate isothermality (ISO), temperature seasonality (TS) and precipitation seasonality (PS).

**Author(s)**

James L. Tsakalos and Martin R. Smith

**References**

Pizarro, M, Hernangómez, D. & Fernández-Avilés G. (2023). climaemet: Climate AEMET Tools. Comprehensive R Archive Network. [doi:10.5281/zenodo.5205573](https://doi.org/10.5281/zenodo.5205573)  
 Walter, H.B., & Lieth, H. (1960). Klimadiagramm-Weltatlas. VEB Gustav Fischer Verlag, Jena.

**See Also**

Download climate data: [ce\\_download\(\)](#)

**Examples**

```
# Step 1. Import the Italian Biome polygon data
# Step 2. Run the download function
# Step 3. Run the extract function
#* See ce_download & ce_extract documentation

# Steps 1, 2 & 3 can be skipped by loading the extracted data (it_data)
data("it_data", package = "climenv")

# Step 4. Visualise the climatic envelope using our custom diagram

# first lets store the default graphics parameters
# we need to make some changes to ensure that the table fits in the plotting
# region.

# Set up plotting parameters
oldpar <- par(mar = c(1.5, 2.2, 1.5, 14) + 0.01)

plot_c(
  it_data, geo_id = "MED",
  l_tcols = c(14.5, 17, 18.5, 19.5, 21)
)

# Restore user options
par(oldpar)

# This output works if you export to a three column width sized image.
```

---

plot_h	<i>plot_h</i>
--------	---------------

---

### Description

Creates a graph using the climate and elevation data which has been extracted for a given location. It accepts the data formatted from the `ce_extract` function.

### Usage

```
plot_h(data, geo_id, col = "red", pch = 19, ...)
```

### Arguments

<code>data</code>	List. Containing climate, elevation and latitude data sets. Structured by <code>ce_extract()</code> .
<code>geo_id</code>	Character. Corresponding to a specific feature contained in the "location" argument (specifically in the "location_g" column).
<code>col</code> , <code>pch</code> , ...	Arguments to control point styling in <code>HoldridgePoints()</code> .

### Value

Returns a base R family of plot. This function uses the **Ternary** package to create a Holdridge simplex plot.

### Author(s)

James L. Tsakalos and Martin R. Smith

### References

Holdridge (1947), Determination of world plant formations from simple climatic data. *Science*, 105:367–368. doi:10.1126/science.105.2727.367

Holdridge (1967), *Life zone ecology*. Tropical Science Center, San José.

Smith, M.R (2017). Ternary: An R Package for Creating Ternary Plots. Comprehensive R Archive Network. doi:10.5281/zenodo.1068996

### See Also

Download climate data: [ce\\_download\(\)](#)

**Examples**

```
# Step 1. Import the Italian Biome polygon data
# Step 2. Run the download function
# Step 3. Run the extract function
#* See ce_download & ce_extract documentation

# Steps 1, 2 & 3 can be skipped by loading the extracted data (it_data)
data("it_data", package = "climenv")

# Step 4. Visualise the climatic envelope using a Holdridge diagram

plot_h(data = it_data, geo_id = "MED")
```

---

plot\_wl

*plot\_wl*


---

**Description**

Creates a graph using the climate and elevation data which has been extracted for a given location. It accepts the data formatted from the `ce_extract` function.

**Usage**

```
plot_wl(data, geo_id, ...)
```

**Arguments**

<code>data</code>	List. Containing climate, elevation and latitude data sets. Structured by <code>ce_extract()</code> .
<code>geo_id</code>	Character. Corresponding to a specific feature contained in the "location" argument (specifically in the "location_g" column).
<code>...</code>	Arguments to control styling in <code>ggclimat_walter_lieth()</code> .

**Value**

Returns a base R family of plot. This function uses the **climaemet** package to create the Walter and Lieth (1960) climatic diagram.

**Author(s)**

James L. Tsakalos and Martin R. Smith

**References**

Pizarro, M, Hernangómez, D. & Fernández-Avilés G. (2023). *climaemet: Climate AEMET Tools*. Comprehensive R Archive Network. doi:10.5281/zenodo.5205573

Walter, H.B., & Lieth, H. (1960). *Klimadiagramm-Weltatlas*. VEB Gustav Fischer Verlag, Jena.



**See Also**

Download climate data: [ce\\_download\(\)](#)

**Examples**

```
# Step 1. Import the Italian Biome polygon data
# Step 2. Run the download function
# Step 3. Run the extract function
#* See ce_download & ce_extract documentation

# Steps 1, 2 & 3 can be skipped by loading the extracted data (it_data)
data("it_data", package = "climenv")

# Step 4. Visualise the climatic envelope using a Walter-Lieth diagram

plot_wl(data = it_data, geo_id = "NEM")
```

---

srtm\_tiles

*SRTM tiles data*

---

**Description**

SRTM tiles data.

**Usage**

```
srtm_tiles
```

**Format**

An object of class SpatialPolygonsDataFrame with 872 rows and 1 columns.

**Details**

Contains tiles to assist downloading from 'geodata'.

**Examples**

```
data("srtm_tiles", package = "climenv")
head(srtm_tiles)
```

---

`worldclim`*Download WorldClim climate data*

---

### Description

`worldclim()` downloads the WorldClim V2.1 climate data for 1970–2000. This includes monthly climate data for minimum, mean, and maximum temperature and precipitation at a resolution of 0.5 minutes of a degree. This function uses the **geodata** to download the worldclim tiles.

### Usage

```
worldclim(output_dir, location, var = "all", ...)
```

### Arguments

<code>output_dir</code>	Character (e.g., <code>"../Desktop/chelsea"</code> ). Pathway to where the data will be stored.
<code>location</code>	A <code>"sp"</code> , <code>"sf"</code> polygon or point object. See <a href="#">sf::st_polygon</a> to make polygons and <a href="#">sf::st_as_sf</a> to make point objects.
<code>var, ...</code>	Arguments to <a href="#">download.file()</a> to control file download. .

### Value

`worldclim()` is called for its side effects and returns NULL. Creates four subfolders named `prec`, `tmax`, `tmin` and `tmean`. Each folder contains 12 GeoTiff (.tif) files, one for each month of the year for the time period 1970–2000. Each of the files are downloaded at a spatial resolution of 0.5 minutes of a degree. The precipitation folder contains average monthly precipitation (mm). The `tmax` folder contains maximum monthly temperature. The `tmin` folder contains minimum monthly temperature. The `tmean` folder contains the average monthly temperature. Temperature values are reported in °C.

### Author(s)

James L. Tsakalos and Martin R. Smith

### References

Fick, S.E. & Hijmans, R.J. (2017). WorldClim 2: new 1km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*. 37, 4302–4315. doi:10.1002/joc.5086

### See Also

Downloading from CHELSA [chelsea\(\)](#) or a more convenient function for downloading other climate and elevation data [ce\\_download\(\)](#).

**Examples**

```
# Download time will depend on the size of the area you wish to access
# climate data for and your internet connection speed.

# Make a polygon file
regents <- sf::st_polygon(
  list(
    cbind(
      "lon" = c(51.537, 51.525, 51.523, 51.530, 51.534, 51.537),
      "lat" = c(-0.150, -0.145, -0.156, -0.167, -0.163, -0.150)
    )
  )
)

# Create temporary file
temp_path <- tempfile()

# Download the WorldClim data
worldclim(
  output_dir = temp_path,
  location = regents
)

# Reset user options
unlink(file.path(temp_path))
```

# Index

## \* datasets

- it\_data, 10
- it\_data\_extended, 11
- it\_pt, 11
- it\_py, 12
- srtm\_tiles, 17

bioclimate, 2

ce\_download, 3

ce\_download(), 6, 7, 9, 14, 15, 17, 18

ce\_extract, 5

chelsa, 7

chelsa(), 4, 18

download.file(), 8, 18

elev, 8

elev(), 4

it\_data, 10

it\_data\_extended, 11

it\_pt, 11

it\_py, 12

plot\_c, 13

plot\_h, 15

plot\_h(), 6

plot\_wl, 16

plot\_wl(), 6

sf::st\_as\_sf, 4, 5, 8, 9, 18

sf::st\_polygon, 4, 5, 8, 9, 18

srtm\_tiles, 17

worldclim, 18

worldclim(), 4, 7