

# Package ‘paisaje’

May 9, 2026

**Type** Package

**Title** Spatial and Environmental Data Tools for Landscape Ecology

**Version** 0.2.0

**Maintainer** Manuel Spínola <mspinola10@gmail.com>

**Description** Provides functions for landscape analysis and data retrieval.

The package allows users to download environmental variables from global datasets (e.g., WorldClim, ESA WorldCover, Nighttime Lights), and to compute spatial and landscape metrics using a hexagonal grid system based on the H3 spatial index. It is useful for ecological modeling, biodiversity studies, and spatial data processing in landscape ecology.

Fick and Hijmans (2017) <doi:10.1002/joc.5086>.

Zanaga et al. (2022) <doi:10.5281/zenodo.7254221>.

Uber Technologies Inc. (2022) ``H3: Hexagonal hierarchical spatial index".

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**URL** <https://manuelspinola.github.io/paisaje/>,  
<https://github.com/ManuelSpinola/paisaje>

**BugReports** <https://github.com/ManuelSpinola/paisaje/issues>

**Imports** dplyr, tidyr, sf, terra, h3jsr, rvest, httr, landscapemetrics,  
spocc, exactextractr, progress, magrittr

**Suggests** ggplot2, rmarkdown, knitr

**VignetteBuilder** knitr

**Depends** R (>= 4.1.0)

**LazyData** true

**Language** en-US

**NeedsCompilation** no

**Author** Manuel Spínola [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-7839-1908>>)

**Repository** CRAN

**Date/Publication** 2026-01-07 23:50:02 UTC

## Contents

calculate_it_metrics . . . . .	2
count_points_in_polygons . . . . .	3
create_cat_esa_10m . . . . .	4
cr_outline_c . . . . .	5
extract_cat_raster . . . . .	6
extract_num_raster . . . . .	7
get_esa_10m . . . . .	8
get_h3_grid . . . . .	9
get_nightlight_data . . . . .	10
get_records . . . . .	11
get_records_by_hexagon . . . . .	12
get_worldclim_future . . . . .	13
get_worldclim_historic . . . . .	15

<b>Index</b>	<b>17</b>
--------------	-----------

---

calculate\_it\_metrics    *Calculate Landscape Complexity Metrics (IT Metrics) per Polygon*

---

### Description

Calculates specified landscape complexity metrics (a subset of Information Theory metrics) from a categorical land-cover raster for each input polygon using `landscapemetrics::sample_lsm()`. This function ensures a safe, alignment-guaranteed join of the results back to the original geometry.

### Usage

```
calculate_it_metrics(landscape_raster, aoi_sf)
```

### Arguments

`landscape_raster`  
A `SpatRaster` object representing the categorical landscape (e.g., LULC).

`aoi_sf`  
An `sf` object containing polygonal geometries (e.g., H3 hexagons) for which the landscape metrics should be calculated.

### Details

This function calculates metrics at the "landscape" level, filtering for "complexity metric" types. The function prioritizes data integrity by adding a temporary `plot_id` column based on row index, which is used by `landscapemetrics`.

Crucially, the function uses `dplyr::left_join` with this `plot_id` for merging the results. This **robust join method** prevents data misalignment that could occur if rows were dropped during metric calculation, which is a significant improvement over the unsafe `cbind` method. The temporary `plot_id` column is removed before the final object is returned.

**Value**

An sf object identical to aoi\_sf, but with new columns appended. The new columns represent the calculated landscape metrics (e.g., lsm\_shdi) with an lsm\_ prefix.

**See Also**

[sample\\_lsm](#) for available metrics.

**Examples**

```
## Not run:  
# Assuming 'lulc' is a SpatRaster and 'hex_grid_sf' is an sf polygon grid  
# metrics_sf <- calculate_it_metrics(lulc, hex_grid_sf)  
# head(metrics_sf)  
  
## End(Not run)
```

---

count\_points\_in\_polygons

*Count Points within Polygons by Species*

---

**Description**

Counts the number of points per species within each polygon. If the points dataset contains a 'species' column, a separate column is created for each species with the counts inside each polygon. Spaces in species names are replaced with underscores for naming columns.

This function is particularly useful in ecological studies where species have different spatial distributions. It accounts for the possibility that some species may not be present in all polygons, producing zero counts in those cases.

**Usage**

```
count_points_in_polygons(points_sf, polygons_sf)
```

**Arguments**

points\_sf      An 'sf' object containing point geometries. Must include a 'species' column.

polygons\_sf    An 'sf' object containing polygon geometries.

**Details**

The function performs a spatial join to count occurrences of each species within each polygon. For species absent in a polygon, the count will be zero. This approach allows for flexible analysis of species distributions across landscape units.

**Value**

An 'sf' object containing the original polygons and additional columns for each species count. Column names follow the format 'species\_name\_count', with spaces replaced by underscores.

**Examples**

```
library(sf)

points_sf <- st_as_sf(data.frame(
  id = 1:6,
  species = c("Panthera onca", "Panthera onca", "Felis catus",
             "Felis catus", "Felis catus", "Panthera leo"),
  x = c(1, 2, 3, 4, 5, 6),
  y = c(1, 2, 3, 4, 5, 6)
), coords = c("x", "y"), crs = 4326)

polygons_sf <- st_as_sf(data.frame(
  id = 1:2,
  geometry = st_sfc(
    st_polygon(list(rbind(c(0,0), c(3,0), c(3,3), c(0,3), c(0,0)))),
    st_polygon(list(rbind(c(3,3), c(6,3), c(6,6), c(3,6), c(3,3))))
  )
), crs = 4326)

result <- count_points_in_polygons(points_sf, polygons_sf)
print(result)
```

---

create_cat_esa_10m	<i>Create Categorical Land Cover Raster from Copernicus ESA World-Cover Data</i>
--------------------	--

---

**Description**

This function takes a 'SpatRaster' object containing Copernicus ESA WorldCover land cover data, reclassifies it into categorical land cover classes based on predefined schemes, and returns the resulting categorical raster.

**Usage**

```
create_cat_esa_10m(land_cover)
```

**Arguments**

land_cover	A 'SpatRaster' object representing the input land cover raster from Copernicus ESA WorldCover. This raster should contain land cover classes as defined by the Copernicus program.
------------	--

**Details**

The function uses a predefined classification scheme for ESA WorldCover data, assigning numeric or categorical values to represent different land cover types. The resulting raster can be used for further spatial analysis or landscape ecology studies.

**Value**

A ‘SpatRaster’ object containing the reclassified categorical land cover raster. Each pixel will have a category corresponding to a defined land cover type.

**References**

Zanaga, D., Van De Kerchove, R., De Keersmaecker, W., et al. (2021). ESA WorldCover 10 m 2020 v100. <https://doi.org/10.5281/zenodo.5571936> Zanaga, D., Van De Kerchove, R., Daems, D., et al. (2022). ESA WorldCover 10 m 2021 v200. <https://doi.org/10.5281/zenodo.7254221> ESA WorldCover project 2020 and 2021. Contains modified Copernicus Sentinel data processed by ESA WorldCover consortium. [ESA WorldCover](#)

**Examples**

```
## Not run:  
# Assuming 'land_cover_raster' is a SpatRaster object from ESA WorldCover  
cat_raster <- create_cat_esa_10m(land_cover_raster)  
  
## End(Not run)
```

---

cr\_outline\_c

*Costa Rica Continental Outline*

---

**Description**

A simplified outline of Costa Rica as an ‘sf’ object.

**Usage**

```
cr_outline_c
```

**Format**

An ‘sf’ object containing polygon geometry of Costa Rica.

**Source**

Adapted from publicly available geographic data.

**Examples**

```
library(sf)
plot(cr_outline_c)
```

---

extract_cat_raster	<i>Calculate Area Proportions for Categorical Raster Classes (Generic)</i>
--------------------	--

---

**Description**

Extracts and calculates the **area proportion** of each categorical class (e.g., LULC) found within each input polygon. This function uses area-weighting to ensure highly accurate, sub-pixel zonal statistics.

**Usage**

```
extract_cat_raster(spat_raster_cat, sf_hex_grid, proportion = TRUE)
```

**Arguments**

spat_raster_cat	A single-layer SpatRaster object containing categorical values.
sf_hex_grid	An sf object containing polygonal geometries. The function will use h3_address if present, otherwise it creates and uses a temporary ID column for joining.
proportion	Logical. If TRUE (default), the output values are the proportion of the polygon area covered by each category (summing to 1 for the covered area). If FALSE, the output is the raw sum of the coverage fraction (area).

**Details**

This function replaces the simplistic, non-area-weighted table() counting method with a robust custom function utilizing dplyr and the coverage\_fraction column from exactextractr. Key features include:

- **Area-Weighted Accuracy:** Uses coverage\_fraction for precise results.
- **NA Filtering:** Excludes NA raster values to prevent a prop\_NaN column.
- **Numerical Ordering:** Sorts the final output columns by category number (e.g., 70 before 80).

**Value**

An sf object identical to sf\_hex\_grid, but with new columns appended for each categorical value found in the raster. Column names follow the pattern <layer\_name>\_prop\_<category\_value>. Columns are **numerically ordered** by the category value.

## Examples

```
## Not run:  
# Assuming 'lulc' is a categorical SpatRaster and 'hex_grid' is an sf polygon grid  
# cat_data_p <- extract_cat_raster(lulc, hex_grid)  
# head(cat_data_p)  
  
## End(Not run)
```

---

extract\_num\_raster      *Extract Area-Weighted Mean from Numeric Raster Stack for Polygons*

---

## Description

Calculates the area-weighted mean value for each layer in a numeric SpatRaster (or single layer) within each polygon feature of an sf object. This function is designed for high-precision zonal statistics of continuous variables (e.g., bioclimatic data).

## Usage

```
extract_num_raster(spat_raster_multi, sf_hex_grid)
```

## Arguments

`spat_raster_multi`      A SpatRaster object from the terra package. Must contain numeric layers (can be a single layer or a stack/brick).  
`sf_hex_grid`      An sf object containing polygonal geometries (e.g., H3 hexagons).

## Details

The function uses `exactextractr::exact_extract` with `fun = "weighted_mean"` and `weights = "area"` to ensure the most accurate sub-pixel summary. A critical security check is implemented before binding columns (`bind_cols`) to prevent data misalignment in case of row count discrepancies between the input features and the extracted results.

## Value

An sf object identical to `sf_hex_grid`, but with new columns appended. The new column names match the original SpatRaster layer names. The values represent the area-weighted mean for that variable within each polygon.

## Examples

```
## Not run:  
# Assuming 'bio' is a SpatRaster stack and 'h7' is an sf hexagon grid  
# bio_p <- extract_num_raster(bio, h7)  
# head(bio_p)  
  
## End(Not run)
```

get\_esa\_10m

*Download ESA WorldCover land cover data***Description**

Downloads ESA WorldCover land cover data at 10 m resolution for a specified area of interest (AOI) and year. Useful for landscape ecology studies, environmental analyses, and habitat mapping.

**Usage**

```
get_esa_10m(aoi_sf, year = 2020, output_folder = NULL)
```

**Arguments**

aoi_sf	‘sf’ An sf object defining the area of interest (AOI). This can be a country, state, or custom boundary.
year	‘numeric’ Year of the land cover data. Available: - 2020: ESA WorldCover 10 m 2020 v100 - 2021: ESA WorldCover 10 m 2021 v200
output_folder	‘character’ Directory where data files will be saved. Default is “.” (current working directory).

**Details**

This function downloads global land-cover raster data produced by the ESA WorldCover project. The downloaded file can be large (hundreds of MB), and processing may take several minutes depending on the AOI size and internet speed.

**\*\*Land-cover classification (ESA WorldCover 10 m v200):\*\***

Value	Class (English)	Categoría (Español)
10	Tree cover	Cobertura arbórea
20	Shrubland	Matorrales
30	Grassland	Pastizales / herbazales
40	Cropland	Tierras de cultivo
50	Built-up	Áreas construidas / urbanas
60	Bare / Sparse vegetation	Vegetación escasa o suelos desnudos
70	Snow and ice	Nieve y hielo permanentes
80	Permanent water bodies	Cuerpos de agua permanentes
90	Herbaceous wetland	Humedales herbáceos
95	Mangroves	Manglares
100	Moss and lichen	Musgos y líquenes

**Value**

‘SpatRaster’ A raster object containing land-cover classification for the specified AOI and year. The raster values correspond to land-cover classes as defined by the ESA WorldCover classification scheme.

**References**

Zanaga, D., Van De Kerchove, R., De Keersmaecker, W., et al. (2021). \*ESA WorldCover 10 m 2020 v100.\* <https://doi.org/10.5281/zenodo.5571936> Zanaga, D., Van De Kerchove, R., Daems, D., et al. (2022). \*ESA WorldCover 10 m 2021 v200.\* <https://doi.org/10.5281/zenodo.7254221>

## Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
get_esa_10m(nc, year = 2021, output_folder = tempdir())
```

---

get\_h3\_grid

*Generate an H3 Hexagonal Grid for an sf Object*

---

## Description

Generates a hexagonal grid of H3 cells at a specified resolution that intersect with a given ‘sf’ object. This is a wrapper for functions from the **h3jsr** package.

## Usage

```
get_h3_grid(sf_object, resolution = 6, expand_factor = 0.1)
```

## Arguments

sf_object	(sf) An sf object defining the area of interest. Must have a valid coordinate reference system (CRS).
resolution	(integer) H3 resolution level (1–16). Default is 6. Lower values produce fewer, larger hexagons (faster processing, coarser grid).
expand_factor	(numeric) Expands bounding box to ensure coverage. Default is 0.1.

## Details

Reducing the resolution (e.g., 5 or 6) can greatly reduce processing time and memory usage, especially for large AOIs. Each decrease in resolution increases the size of individual hexagons exponentially.

## Value

(sf) An sf object containing the hexagonal grid polygons covering the input area. Each polygon represents an H3 cell at the specified resolution, with a column containing the H3 index.

## Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package="sf"))
h3_grid_sf <- get_h3_grid(nc, resolution = 6)
```

---

get\_nightlight\_data     *Download and Retrieve Nightlight Data*

---

## Description

Downloads nightlight data from the Earth Observation Group's website. It scrapes the website to locate and download the latest available nightlight dataset for the specified year and month.

## Usage

```
get_nightlight_data(  
    year,  
    month,  
    version = "v10",  
    destination_dir = NULL,  
    timeout = 1200  
)
```

## Arguments

year	'numeric' o 'character' The year for which to download nightlight data (e.g., 2020).
month	'numeric' o 'character' Month of the year (1–12). Will be formatted as two digits (e.g., "03" for March).
version	'character' Nightlight data version. Default is "v10".
destination_dir	'character' Directory where the downloaded '.tif' file will be saved. Default is the current working directory ".".
timeout	'numeric' Timeout in seconds for the download. Default is '1200' seconds.

## Details

The function constructs the appropriate URL for the specified year, month, and data version, then scrapes the directory listing to locate the latest available '.tif' file. It downloads and saves the file to the 'destination\_dir'. This function is useful for retrieving nightlight data for studies involving human activity, urbanization, and environmental monitoring.

## Value

'character' o 'NULL' Path to the downloaded '.tif' file. Returns 'NULL' if no file was found or if an error occurred.

**Examples**

```
# Download nightlight data for March 2021
file_path <- get_nightlight_data(2021, 3)
print(file_path)
```

get\_records

*Query Species Occurrence Records within an Area of Interest (AOI)***Description**

Downloads species occurrence records from providers (e.g., GBIF) using the spocc package, filtering the initial query by the exact polygonal boundary of the Area of Interest (AOI) for maximum efficiency and precision.

**Usage**

```
get_records(
  species,
  aoi_sf,
  providers = NULL,
  limit = 500,
  remove_duplicates = FALSE,
  date = NULL
)
```

**Arguments**

species	Character string specifying the species name to query (e.g., "Puma concolor").
aoi_sf	An sf object defining the Area of Interest (AOI). Its CRS will be transformed to WGS84 (EPSG: 4326) before query.
providers	Character vector of data providers to query (e.g., "gbif", "inat"). If NULL (default), all available providers are used.
limit	Numeric. The maximum number of records to retrieve per provider. Default is 500.
remove_duplicates	Logical. If TRUE, records with identical longitude and latitude are removed using <code>dplyr::distinct()</code> . Default is FALSE.
date	Character vector specifying a date range (e.g., <code>c('2000-01-01', '2020-12-31')</code> ).

**Details**

The function transforms the aoi\_sf polygon into a WKT string, which is used in the `spocc::occ` geometry argument. This method is more efficient than querying by the rectangular bounding box, as it reduces the number of irrelevant records downloaded. Final spatial filtering is performed using `sf::st_intersection` to ensure strict containment.

**Value**

An sf object of points containing the filtered occurrence records, with geometry confirmed to fall strictly within the aoi\_sf boundary.

**Examples**

```
## Not run:
# Assuming aoi_sf is a valid sf polygon
# puma_records <- get_records("Puma concolor", aoi_sf, providers = "gbif", limit = 1000)
# head(puma_records)

## End(Not run)
```

---

```
get_records_by_hexagon
```

*Retrieve species records aggregated by H3 hexagons*

---

**Description**

Downloads species occurrence data within a specified Area of Interest (AOI) and aggregates these records into H3 hexagonal grid cells at a given resolution. Returns an 'sf' object with one polygon per hexagon and columns containing species occurrence counts.

**Usage**

```
get_records_by_hexagon(
  species, aoi_sf, res = 6,
  providers = NULL, remove_duplicates = FALSE,
  date = NULL, expand_factor = 0.1, limit = 500
)
```

**Arguments**

species	character vector. Species names to query.
aoi_sf	sf object. Area of Interest polygon.
res	integer. H3 resolution level (1–16). Default: 6.
providers	character vector. Data providers to query. Default: NULL (all).
remove_duplicates	logical. Remove duplicate records. Default: FALSE.
date	character vector of length two. Start and end dates for filtering records.
expand_factor	numeric. Expand AOI bounding box. Default: 0.1.
limit	integer. Maximum number of occurrence records per species. Default: 500.

## Details

This function is useful for spatial biodiversity analyses where data should be aggregated into a uniform spatial grid. The H3 grid system enables multi-resolution analysis and efficient spatial summarization of point occurrence data.

## Value

sf object. H3 hex grid with species occurrence counts.

## Examples

```
library(sf)
nc <- sf::st_read(system.file("shape/nc.shp", package="sf"))
hex_counts <- get_records_by_hexagon(
  species = c("Lynx rufus"),
  aoi_sf = nc,
  res = 6,
  limit = 200
)
print(hex_counts)
```

---

get\_worldclim\_future *Download and process future environmental variables from WorldClim v2.1*

---

## Description

Downloads future climate data from WorldClim based on CMIP6 climate models and SSP scenarios. The data can be retrieved at various spatial resolutions and optionally clipped to a specified area of interest (AOI).

## Usage

```
get_worldclim_future(
  var = "bioc",
  res = "30s",
  scenario = "585",
  time_range = "2021-2040",
  gcm = "ACCESS-CM2",
  aoi = NULL,
  retries = 3,
  timeout = 300,
  destination_dir = NULL
)
```

**Arguments**

<code>var</code>	<p>character Climate variable to download. Options:</p> <ul style="list-style-type: none"> <li>• "bioc" — Bioclimatic variables (19 variables)</li> <li>• "prec" — Precipitation</li> <li>• "tavg" — Average temperature</li> <li>• "tmin" — Minimum temperature</li> <li>• "tmax" — Maximum temperature</li> </ul> <p>Default is "bioc".</p>
<code>res</code>	<p>character Spatial resolution of the data. Options:</p> <ul style="list-style-type: none"> <li>• "30s" — ~1 km (30 arc-seconds)</li> <li>• "2.5m" — ~5 km (2.5 arc-minutes)</li> <li>• "5m" — ~10 km (5 arc-minutes)</li> <li>• "10m" — ~20 km (10 arc-minutes)</li> </ul> <p>Default is "30s".</p>
<code>scenario</code>	<p>character SSP scenario. Options:</p> <ul style="list-style-type: none"> <li>• "126" — SSP1-2.6 (low emissions)</li> <li>• "245" — SSP2-4.5 (intermediate emissions)</li> <li>• "370" — SSP3-7.0 (high emissions)</li> <li>• "585" — SSP5-8.5 (very high emissions)</li> </ul> <p>Default is "585".</p>
<code>time_range</code>	<p>character Time period. Options:</p> <ul style="list-style-type: none"> <li>• "2021-2040"</li> <li>• "2041-2060"</li> <li>• "2061-2080"</li> <li>• "2081-2100"</li> </ul> <p>Default is "2021-2040".</p>
<code>gcm</code>	<p>character General Circulation Model. Options: "ACCESS-CM2", "ACCESS-ESM1-5", "AWI-CM-1-1-MR", "BCC-CSM2-MR", "CanESM5", "CanESM5-CanOE", "CMCC-ESM2", "CNRM-CM6-1", "CNRM-CM6-1-HR", "CNRM-ESM2-1", "EC-Earth3-Veg", "EC-Earth3-Veg-LR", "FIO-ESM-2-0", "GFDL-ESM4", "GISS-E2-1-G", "GISS-E2-1-H", "HadGEM3-GC31-LL", "INM-CM4-8", "INM-CM5-0", "IPSL-CM6A-LR", "MIROC-ES2L", "MIROC6", "MPI-ESM1-2-HR", "MPI-ESM1-2-LR", "MRI-ESM2-0", "UKESM1-0-LL". Default is "ACCESS-CM2".</p>
<code>aoi</code>	<p>sf or SpatRaster Optional area of interest to clip the data. Default is NULL (no clipping).</p>
<code>retries</code>	<p>integer Number of attempts to retry download in case of failure. Default is 3.</p>
<code>timeout</code>	<p>numeric Download timeout in seconds. Default is 300.</p>
<code>destination_dir</code>	<p>character Directory where downloaded data will be stored. Default is NULL (uses a temporary directory).</p>

**Value**

SpatRaster object containing the selected climate variables, optionally clipped to the specified AOI.

**References**

Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, 37(12), 4302–4315. doi:10.1002/joc.5086

**Examples**

```
nc <- sf::st_read(system.file("shape/nc.shp", package = "sf"))
nc <- sf::st_transform(nc, crs = 4326)

climate_future <- paisaje::get_worldclim_future(
  var = "tmin", res = "10m", scenario = "585",
  time_range = "2021-2040", gcm = "ACCESS-CM2", aoi = nc
)
```

---

get\_worldclim\_historic

*Descargar y procesar variables climáticas históricas de WorldClim v2.1*

---

**Description**

Descarga datos climáticos históricos de WorldClim v2.1 y los procesa según los parámetros especificados. Soporta múltiples variables climáticas y resoluciones espaciales. Opcionalmente recorta los datos a un área de interés (AOI).

**Usage**

```
get_worldclim_historic(
  var = "bio",
  res = 10,
  aoi = NULL,
  retries = 3,
  timeout = 300,
  destination_dir = NULL
)
```

**Arguments**

var                      Character. Variable climática a descargar. Opciones:

- "bio" — Variables bioclimáticas.
- "tavg" — Temperatura media.

- "tmin" — Temperatura mínima.
- "tmax" — Temperatura máxima.
- "prec" — Precipitación.
- "srad" — Radiación solar.
- "wind" — Velocidad del viento.
- "vapr" — Presión de vapor.

Por defecto: "bio".

**res** Numeric. Resolución espacial en minutos de arco. Valores válidos: '0.5', '2.5', '5', '10'. Estos valores se mapean internamente a cadenas aceptadas por WorldClim:

- 0.5 → "30s"
- 2.5 → "2.5m"
- 5 → "5m"
- 10 → "10m"

Por defecto: '10'.

**aoi** sf o SpatRaster opcional. Área de interés para recortar los datos.

**retries** Integer. Número de intentos de descarga en caso de fallo. Por defecto: '3'.

**timeout** Numeric. Tiempo máximo de descarga en segundos. Por defecto: '300'.

**destination\_dir**

Character. Carpeta donde guardar los datos descargados. Si NULL, se usa un directorio temporal.

### Value

Un objeto 'SpatRaster' con las variables climáticas históricas. Si se especifica 'aoi', los datos se recortan a esa área.

### References

Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, 37(12), 4302–4315. doi:10.1002/joc.5086

### Examples

```
nc <- sf::st_read(system.file("shape/nc.shp", package="sf"))
nc <- sf::st_transform(nc, crs = 4326)

climate_historic <- get_worldclim_historic(
  var = "tmin",
  res = 5,
  aoi = nc
)
```

# Index

## \* datasets

- cr\_outline\_c, [5](#)
  
- calculate\_it\_metrics, [2](#)
- count\_points\_in\_polygons, [3](#)
- cr\_outline\_c, [5](#)
- create\_cat\_esa\_10m, [4](#)
  
- extract\_cat\_raster, [6](#)
- extract\_num\_raster, [7](#)
  
- get\_esa\_10m, [8](#)
- get\_h3\_grid, [9](#)
- get\_nightlight\_data, [10](#)
- get\_records, [11](#)
- get\_records\_by\_hexagon, [12](#)
- get\_worldclim\_future, [13](#)
- get\_worldclim\_historic, [15](#)
  
- sample\_lsm, [3](#)