

Package: valuate (via r-universe)

October 26, 2026

Type Package

Title Caja de herramientas para la realización de valuaciones masivas automatizadas

Version 0.2.0

Description Provee funciones para facilitar el proceso de valuación masiva de inmuebles, siguiendo el desarrollo de Eguino y Carranza (2025) <doi:10.18235/0013669>. Permite crear variables territoriales que informan sobre la distancia a diferentes hitos o las características del entorno, homogeneizar valores para descontar los efectos de características intrínsecas de las observaciones, identificar observaciones espacialmente atípicas, entrenar modelos de aprendizaje computacional estimando su error fuera de la muestra y simular escenarios para conocer la magnitud y extensión geográfica de diferentes externalidades sobre el precio de los inmuebles. Puede ser también de utilidad para la predicción de la distribución espacial de cualquier tipo de fenómeno geográfico.

URL <https://r-valuate.github.io/valuate/>,
<https://github.com/r-valuate/valuate>,
<https://r-valuate.r-universe.dev/valuate>

BugReports <https://github.com/r-valuate/valuate/issues>

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports dplyr, sf, sp, spdep, terra, raster, fasterize, viridis, caret, quantregForest, parallel, doParallel, data.table, kableExtra, spatialreg, stringr, utils, graphics, stats, tools, methods

RoxygenNote 7.3.2

License GPL (>= 3)

Suggests nominatimlite, osmdata, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs libabsl-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev make libharfbuzz-dev libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

Date/Publication 2026-10-26 22:04:42 UTC

RemoteUrl <https://github.com/r-valuate/valuate>

RemoteRef HEAD

RemoteSha e84055ae48e61e422083c449b50a4b90f0d0e1dc

Contents

calcular_dist	2
calcular_entorno	3
calcular_raster	4
calcular_vut	5
dat	7
datos_disponibles	7
descargar_datos	8
eliminar_outliers	9
entrenar_modelo	10
resultado	12
simular_escenario	14
terrenos	14
Index	16

calcular_dist	<i>Calculo de variables de distancia</i>
---------------	--

Description

Esta funcion devuelve un raster con las distancias euclidianas, en metros, desde cada pixel hacia el objeto vectorial mas cercano.

Usage

```
calcular_dist(area, objeto, dim, nombre)
```

Arguments

area	Poligono que indica el area sobre la cual se ejecutara el calculo.
objeto	Objeto espacial "sf" contra el que se calcularan las distancias.
dim	Tamano de cada pixel del raster resultante, en metros.
nombre	Nombre de la variable resultante. El raster se guarda ademas en el directorio de trabajo con este nombre y la extension ".tif".

Value

Un objeto 'SpatRaster' (libreria 'terra'). Se genera un raster que informa, para cada pixel, sobre la distancia (en metros lineales) al objeto mas cercano.

Author(s)

Juan Pablo Carranza

Examples

```
## Not run:
# Definimos el area de estudio
area_de_estudio <- nominatimlite::geo_lite_sf(address = "La Plata, Argentina", points_only = FALSE)
bbox = sf::st_transform(area_de_estudio, 4326) |> sf::st_bbox(bbox)

# Descargamos las vias principales
vias_prim <- osmdata::opq(bbox) |>
  osmdata::add_osm_feature(key = "highway", value = "secondary") |>
  osmdata::osmdata_sf()
vias_prim <- vias_prim$osm_lines

# Calculamos un raster con las distancias a la via principal mas cercana
vias <- calcular_dist(area = area_de_estudio,
                     objeto = vias_prim,
                     dim = 50,
                     nombre = "vias")

## End(Not run)
```

calcular_entorno

Calculo de variables de entorno

Description

Esta funcion devuelve un raster que resume las características del vecindario a partir de capas geográficas vectoriales de poligonos o de puntos.

Usage

```
calcular_entorno(area, objeto, dim, ext, nombre)
```

Arguments

area	Poligono que indica el area sobre la cual se ejecutara el calculo.
objeto	Objeto espacial "sf" con geometria del tipo "POLYGON" o "MULTIPOLYGON" sobre el que se calcularan las características del entorno.
dim	Tamano de cada pixel del raster resultante, en metros.
ext	Extension del entorno o radio que define el vecindario, en metros lineales.

nombre Nombre de la variable resultante. El raster se guarda ademas en el directorio de trabajo con este nombre y la extension ".tif".

Value

Un objeto 'SpatRaster' (libreria 'terra'). Cuando se le provee un objeto espacial "sf" con geometria "POLYGON" o "MULTIPOLYGON" la funcion permite calcular variables independientes que resumen las características del vecindario, indicando la proporcion del espacio que se encuentra cubierto por los poligonos de interes. Por ejemplo, permite calcular el porcentaje del vecindario que se encuentra cubierto por plazas o parques. Cuando se le provee un objeto espacial "sf" con geometria "POINT" la funcion permite calcular variables independientes que resumen las características del vecindario, indicando la cantidad de puntos de interes que se encuentran dentro del entorno definido por el usuario. Por ejemplo, permite calcular la cantidad de negocios u oficinas que se encuentran dentro de un radio de "x" metros de cada pixel.

Author(s)

Juan Pablo Carranza

Examples

```
## Not run:
# Definimos el area de estudio
area_de_estudio <- nominatimlite::geo_lite_sf(address = "La Plata, Argentina", points_only = FALSE)
bbox = sf::st_transform(area_de_estudio, 4326) |> sf::st_bbox(bbox)

# Descargamos los parques que se encuentran dentro del area de estudio
plazas <- osmdata::opq(bbox) |>
  osmdata::add_osm_feature(key = "leisure", value = "park") |>
  osmdata::osmdata_sf()
plazas <- plazas$osm_polygons

# Calculamos un raster que resume, para cada pixel, las características del entorno
plazas_entorno <- calcular_entorno(area = area_de_estudio,
                                   ext = 500,
                                   objeto = plazas,
                                   dim = 50,
                                   nombre = "plazas_entorno")

## End(Not run)
```

calcular_raster

Calculo de variables raster

Description

Esta funcion devuelve un raster que resume el calculo realizado a partir de otro raster, incluyendo fuentes de datos con diferente proyeccion y resolucion espacial.

Usage

```
calcular_raster(raster, area, dim, entorno = 0, nombre)
```

Arguments

raster	Raster original sobre el que se pretende realizar el calculo.
area	Area sobre la que se desea realizar el calculo.
dim	Tamano de cada pixel del raster resultante, en metros.
entorno	Distancia sobre la que se pretende resumir la informacion del raster original. Default = 0. En todo otro caso se calcula el promedio de los valores de los pixeles en un area circular con un radio igual a la distancia establecida en este parametro.
nombre	Nombre de la variable resultante. El raster se guarda ademas en el directorio de trabajo con este nombre y la extension ".tif".

Value

Un objeto 'SpatRaster' (libreria 'terra'). La funcion permite readecuar las resoluciones y proyecciones de diferentes rasters a las utilizadas en el proyecto. Ademas, el parametro "entorno" permite realizar un calculo del promedio de la variable bajo analisis en el vecindario definido por el usuario.

Author(s)

Juan Pablo Carranza

Examples

```
## Not run:
original <- terra::rast(descargar_datos("edificaciones.tif"))
area_de_estudio <- sf::st_read(system.file("extdata/area.gpkg", package = "valuate"))
edificaciones <- calcular_raster(original, area_de_estudio, dim = 50, entorno = 100,
                                nombre = "edificaciones")

## End(Not run)
```

calcular_vut

Homogeneizacion del precio de mercado de los inmuebles

Description

Esta funcion estima el valor de las observaciones muestrales expresado en terminos homogeneos o comparables.

Usage

```
calcular_vut(df, index, dependiente, independientes, dist)
```

Arguments

df	Objeto 'sf' con geometria 'POINT'. Muestra georreferenciada con precios de la tierra cuyo valor se pretende expresar en terminos homogeneos.
index	Identificador unico para cada variable.
dependiente	Nombre de la variable dependiente, aquella que contiene el precio por metro cuadrado.
independientes	Vector con los nombres de las columnas a utilizarse como variables independientes, y el valor de referencia correspondiente a un "lote tipico". Si las variables son numericas se debe indicar si el ajuste a un lote tipico debe tomar como referencia la media o la mediana de dicha variable. En el caso de ser variables categoricas, se debe indicar el nombre de la categoria a tomarse como referencia. Ver ejemplo abajo.
dist	Distancia que define el vecindario sobre el que se realizara la correccion por dependencia espacial.

Value

Una lista con dos elementos: datos, la muestra original (objeto 'sf') con dos columnas nuevas, vut (precio homogeneizado) y coef (coeficiente de homogeneizacion aplicado); y efectos, una tabla con los efectos estadisticamente significativos de cada variable. La lista se devuelve de forma invisible, y la tabla de efectos se imprime en la consola. La funcion estima el precio por metro cuadrado homogeneizado, siguiendo a Carranza (2024). Toma como input un objeto espacial 'sf' y ajusta coeficientes de correccion segun diferentes variables definidas por el usuario para reexpresar el precio en termino de un 'lote tipico' para el area de estudio. Se informa, ademas, el parametro de ajuste estimado por el modelo lineal para cada una de las variables independientes involucradas en el proceso de homogeneizacion.

Author(s)

Juan Pablo Carranza

Examples

```
## Not run:
# Cargamos los datos correspondientes a una muestra de 1000 terrenos.
data("terrenos", package = "valuate")

# Se define como "lote tipico" a aquel que se corresponde a una operacion de
# venta, realizada con escritura, y de una superficie igual a la mediana
# de los lotes de la ciudad.
# Se indica que la variable que informa sobre el precio por metro cuadrado
# es "valor_m2".
# Se define al vecindario en base a una distancia euclidiana de 500 metros.

vut <- calcular_vut(df = terrenos,
  index = "id",
  independientes = c("tipo_valor" = "Venta",
    "superficie" = "mediana",
    "situacion_juridica" = "Con Escritura"),
```

```

                                dependiente = "valor_m2",
                                dist = 500)

# Muestra con el precio homogeneizado
vut$datos

# Efectos estimados
vut$efectos

## End(Not run)

```

dat	<i>Conjunto de datos 'datos'</i>
-----	----------------------------------

Description

Este dataset contiene valores de utilidad territorial ('vut') y su ubicación geográfica.

Usage

```
dat
```

Format

Un data frame con las siguientes columnas:

vut Valor de utilidad territorial del punto.

geometry Coordenadas geográficas del punto.

Source

Datos geoespaciales procesados.

datos_disponibles	<i>Datos de ejemplo disponibles</i>
-------------------	-------------------------------------

Description

Lista los rasters de ejemplo que pueden descargarse con `descargar_datos()`.

Usage

```
datos_disponibles()
```

Value

Un vector con los nombres de los archivos disponibles.

Author(s)

Juan Pablo Carranza

Examples

```
datos_disponibles()
```

descargar_datos

Descarga de los datos de ejemplo

Description

Descarga los rasters de ejemplo de la Ciudad de Medellin (Colombia) que se utilizan en los ejemplos de la documentacion y en el README. Por su tamano, estos archivos no se incluyen en el paquete: se descargan desde el repositorio del proyecto en GitHub y se guardan en una carpeta de cache del usuario, por lo que solo se descargan la primera vez.

Usage

```
descargar_datos(  
  archivos = datos_disponibles(),  
  directorio = tools::R_user_dir("valuate", "cache"),  
  sobrescribir = FALSE  
)
```

Arguments

archivos	Nombres de los archivos a descargar. Por defecto, todos los disponibles (ver <code>datos_disponibles()</code>).
directorio	Carpeta en la que se guardan los archivos. Por defecto, la carpeta de cache del paquete, que se obtiene con <code>tools::R_user_dir("valuate", "cache")</code> .
sobrescribir	Logical. Si es 'TRUE', los archivos se descargan nuevamente aunque ya existan.

Value

Un vector con las rutas a los archivos descargados, que puede pasarse directamente a `terra::rast()` o al argumento independientes de `entrenar_modelo()`.

Author(s)

Juan Pablo Carranza

Examples

```
datos_disponibles()
## Not run:
rutas <- descargar_datos(c("dist_tren.tif", "entorno_comercios.tif"))
terra::rast(rutas)

## End(Not run)
```

eliminar_outliers	<i>Identificacion de valores espacialmente atipicos</i>
-------------------	---

Description

Esta funcion permite eliminar observaciones cuyo valor es muy diferente a los observados en el entorno. Permite identificar outliers espaciales mediante el calculo del indice de Moran local, o mediante el promedio observado en el entorno ponderado por la distancia.

Usage

```
eliminar_outliers(df, dist, variable, umbral, moran)
```

Arguments

df	Dataframe sobre el que se realizara el calculo. Debe ser un objeto de la libreria 'sf' con geometria 'POINT'.
dist	Distancia euclidiana que define la extension del vecindario en el que se compararan los valores.
variable	Nombre de la variable sobre la que se realizara el calculo.
umbral	Umbral para caracterizar a una observacion como atipica en relacion a lo observado en el vecindario. Por ejemplo, un umbral igual a 0.1 implicara que se caracterizaran como outliers a todas aquellas observaciones cuyo valor este por encima o por debajo del observado en el vecindario en una magnitud igual a +/- 10%.
moran	Logical. Si asume el valor 'TRUE' se procede a identificar outliers mediante el calculo del indice de Moran local. En este caso, el parametro 'umbral' queda sin efecto.

Value

Un objeto 'sf' con las observaciones de la muestra original, sin las identificadas como espacialmente atipicas. La funcion permite identificar valores que son muy diferentes a los observados en el entorno. Para ello, compara el valor de cada observacion con el promedio del entorno (ponderado por la distancia, mediante la construccion de una matriz de pesos espaciales). Tambien permite identificar outliers espaciales mediante el calculo del indice de Moran local, siguiendo a Anselin (1995).

Author(s)

Juan Pablo Carranza

References

Anselin, L. (1995). Local Indicators of Spatial Association - LISA. *Geographical Analysis*, **27**(2), 93-115. doi:10.1111/j.15384632.1995.tb00338.x. Disponible en: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1538-4632.1995.tb00338.x>.

Examples

```
# Cargamos la base de datos y tomamos las primeras 1500 observaciones
data("resultado", package = "valuate")
muestra <- resultado[1:1500, ]

# Se define la estimacion de observaciones atipicas a partir de la variable 'vut'
# La distancia para el calculo es de 50 metros lineales. Esto define la extension del vecindario.
# El umbral para considerar a una observacion como atipica se define igual a 0.1 (+/- 10%)
# No se realiza el calculo mediante el indice de Moran local.
depurados <- eliminar_outliers(df = muestra,
                              dist = 50,
                              variable = "vut",
                              umbral = 0.1,
                              moran = FALSE)

# Si, en cambio, se desea identificar valores atipicos mediante
# el calculo del indice de Moran local:
depurados_moran <- eliminar_outliers(df = muestra,
                                     dist = 50,
                                     variable = "vut",
                                     moran = TRUE)
```

entrenar_modelo

Entrenamiento de modelos y prediccion del valor

Description

Esta funcion permite entrenar modelos y cuantificar su error en la prediccion del valor de los inmuebles (o cualquier otra variable territorial que se este analizando).

Usage

```
entrenar_modelo(
  df,
  dependiente,
  independientes,
  modelo = "qrf",
  umbral = 0.3,
  eliminar = 0.4
)
```

Arguments

<code>df</code>	Dataframe sobre el que se realizara el calculo. Debe ser un objeto de la libreria 'sf' con geometria 'POINT'.
<code>dependiente</code>	Nombre de la variable a estimar o variable dependiente.
<code>independientes</code>	Vector con los nombres de los rasters que contienen informacion sobre las variables independientes, generadas a partir de las funciones <code>calcular_dist()</code> , <code>calcular_raster()</code> o <code>calcular_entorno()</code> .
<code>modelo</code>	Nombre del modelo a entrenar. Se corresponde con las abreviaciones previstas en la libreria 'caret' (Kuhn, 2008) (ver https://topepo.github.io/caret/train-models-by-tag.html). El valor por default ("qrf") se corresponde al modelo Quantile Regression Forest (Meinshausen, 2006).
<code>umbral</code>	Valor que define el error maximo permitido en la estimacion. La funcion evalua la precision fuera de la muestra mediante un proceso de validacion cruzada en 10 grupos. Para cada dato en la muestra se registra el error fuera de la muestra. Si el nivel de error agregado es superior al umbral definido, la funcion procede a eliminar los datos con un nivel de error mayor o igual al definido por el parametro 'eliminar' (ver abajo).
<code>eliminar</code>	Valor que define el nivel de error a partir del cual se depurara la muestra, siempre que el error general de estimacion se encuentre por encima del definido por el parametro 'umbral'.

Value

Una lista (devuelta de forma invisible) con cuatro elementos: prediccion, un objeto 'SpatRaster' con la prediccion espacial; modelo, el modelo entrenado; utilizados, los datos que cumplieron con el umbral de error; y eliminados, los datos descartados por errores altos. Ademàs, la prediccion y el modelo se guardan en el directorio de trabajo con los nombres 'vut.tif' y 'modelo.rda'.

Esta funcion implementa un flujo completo de entrenamiento, evaluacion y ajuste de un modelo supervisado para la estimacion de una variable dependiente a partir de un conjunto de variables independientes en formato raster. Especificamente, esta disenada para trabajar con datos espaciales almacenados en un dataframe con geometria de tipo 'POINT', usando herramientas de las librerias 'sf' y 'terra'. El proceso se desarrolla en los siguientes pasos principales:

1. Preprocesamiento de datos: Los rasters de las variables independientes se combinan con el dataframe espacial mediante la extraccion de valores correspondientes a los puntos de la geometria. Los datos se depuran eliminando filas con valores faltantes.
2. Configuracion del modelo: Se define un modelo a traves de la libreria 'caret', con soporte para una amplia variedad de algoritmos. El modelo predeterminado es Quantile Regression Forest (QRF), aunque se pueden especificar otros metodos soportados por 'caret'.
3. Entrenamiento iterativo: La funcion utiliza validacion cruzada de 10 particiones para evaluar el error medio absoluto porcentual (MAPE) fuera de la muestra. Si el MAPE promedio supera el umbral especificado, se eliminan observaciones cuyo MAPE individual sea igual o superior al 40%, y el modelo se entrena nuevamente con los datos restantes. Este proceso se repite hasta que el MAPE promedio cumpla con el umbral definido.
4. Generacion de prediccion espacial: Una vez finalizado el entrenamiento, el modelo entrenado se utiliza para interpolar las predicciones sobre los rasters de entrada, generando un nuevo raster de salida que se guarda en el directorio de trabajo con el nombre 'vut.tif'.

5. Salida de datos: La funcion devuelve los datos que cumplieron con el umbral de error (utilizados) y los descartados por errores altos (eliminados).

La funcion esta optimizada para aplicaciones que requieren analisis espaciales precisos y esta disenada para integrarse en flujos de trabajo geoespaciales complejos. Ademas, emplea paralelizacion para acelerar el proceso de entrenamiento en maquinas con multiples nucleos. El diseno de esta funcion esta orientado a contextos donde es crucial garantizar una alta precision en la estimacion, especialmente en aplicaciones que involucran modelizacion espacial o analisis urbano. Ademas, facilita la integracion con flujos de trabajo basados en datos geoespaciales mediante las librerias 'sf' y 'raster'.

Author(s)

Juan Pablo Carranza

References

- Kuhn, M. (2008). "Building Predictive Models in R Using the caret Package". *Journal of Statistical Software*, **28**(5), 93-115. <https://doi.org/10.18637/jss.v028.i05> Disponible en: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1538-4632.1995.tb00338.x>.
- Meinshausen, N. (2006) "Quantile Regression Forests". *Journal of Machine Learning Research*, **7**. Disponible en: <https://jmlr.csail.mit.edu/papers/v7/>.

Examples

```
## Not run:
# A continuacion se presenta un ejemplo para la Ciudad de Medellin (Colombia), con solo 500 datos.
data("datos", package = "valuate")
rasters <- c("dist_basura.tif", "dist_industria.tif", "dist_plazas_parques.tif",
            "dist_tren.tif", "dist_vias_autopista.tif", "dist_vias_prim.tif",
            "dist_vias_sec.tif", "entorno_altura.tif", "entorno_plazas_parques.tif",
            "entorno_comercios.tif", "entorno_edificaciones.tif", "entorno_hoteles.tif")
modelo <- entrenar_modelo(df = dat,
                        dependiente = "vut",
                        independientes = descargar_datos(rasters),
                        modelo = "grf",
                        umbral = 0.3)

modelo$prediccion
modelo$utilizados

## End(Not run)
```

resultado

Conjunto de datos 'resultado'

Description

Este dataset contiene información detallada sobre valores inmobiliarios, características del entorno y distancias a distintos servicios.

Usage

resultado

Format

Un data frame con las siguientes columnas:

id Identificador único del terreno.
valor Valor total del terreno en la moneda especificada.
moneda Moneda en la que está expresado el valor del terreno.
tipo_valor Tipo de valor registrado.
superficie Superficie total del terreno en metros cuadrados.
situacion_juridica Estado legal del terreno.
ID Identificador adicional para cruces de datos.
area_plazas Porcentaje del entorno cubierto por plazas.
comercios_entorno Cantidad de comercios en el entorno.
d_bancos Distancia a bancos más cercanos.
d_barrio_cerrado Distancia al barrio cerrado más cercano.
d_escuela Distancia a la escuela más cercana.
d_hospital Distancia al hospital más cercano.
d_plazas Distancia a la plaza más cercana.
d_policia Distancia a la comisaría más cercana.
d_rio Distancia al río más cercano.
d_universidad Distancia a la universidad más cercana.
d_vias_primarias Distancia a la vía primaria más cercana.
d_vias_secundarias Distancia a la vía secundaria más cercana.
valor_m2 Valor por metro cuadrado del terreno.
coef Coeficiente ajustado.
vut Valor de utilidad territorial ajustado.
geometry Coordenadas geográficas del terreno.

Source

Datos derivados del procesamiento espacial y económico de terrenos.

simular_escenario	<i>Simulacion de escenarios y estimacion de externalidades espaciales</i>
-------------------	---

Description

A partir de la prediccion de la distribucion espacial de una variable, la funcion permite estimar escenarios a partir de la simulacion de alteraciones en alguna (o algunas) de las variables independientes.

Usage

```
simular_escenario(modelo, independientes, original)
```

Arguments

modelo	Archivo .rda guardado en el directorio de trabajo por la funcion entrenar_modelo().
independientes	Vector con los nombres de las variables independientes. Debe tener los mismos nombre que los utilizados en la funcion entrenar_modelo(), solo que admite que los rasters tengan una distribucion espacial diferente, incorporando modificaciones simuladas a la nueva prediccion.
original	Resultado del modelo original. Es un archivo .tif guardado en el directorio de trabajo por la funcion entrenar_modelo().

Value

Un objeto 'SpatRaster' (libreria 'terra'), que tambien se guarda en el directorio de trabajo con el nombre 'impacto.tif'. La funcion devuelve un raster que informa sobre el impacto en la variable dependiente de las simulaciones realizadas en las variables independientes. Permite estimar la magnitud y el alcance territorial de las modificaciones simuladas sobre la variable de estudio.

Author(s)

Juan Pablo Carranza

terrenos	<i>Conjunto de datos de terrenos</i>
----------	--------------------------------------

Description

Este dataset contiene información sobre terrenos en venta, incluyendo el precio, moneda, tipo de valor, superficie y situación jurídica.

Usage

```
terrenos
```

Format

Un data frame con las siguientes columnas:

id Identificador único del terreno.

valor Valor total del terreno en la moneda especificada.

moneda Moneda en la que está expresado el valor del terreno (por ejemplo, "Dólar (USD)").

tipo_valor Tipo de valor registrado (por ejemplo, "Oferta").

superficie Superficie total del terreno en metros cuadrados.

situacion_juridica Estado legal del terreno (por ejemplo, "Con Escritura", "Pre venta").

valor_m2 Valor por metro cuadrado del terreno.

geom Coordenadas del terreno en el sistema de referencia utilizado.

Source

Datos generados a partir de anuncios de terrenos en venta.

Index

* datasets

- dat, [7](#)
- resultado, [12](#)
- terrenos, [14](#)

- calcular_dist, [2](#)
- calcular_entorno, [3](#)
- calcular_raster, [4](#)
- calcular_vut, [5](#)

- dat, [7](#)
- datos_disponibles, [7](#)
- descargar_datos, [8](#)

- eliminar_outliers, [9](#)
- entrenar_modelo, [10](#)

- resultado, [12](#)

- simular_escenario, [14](#)

- terrenos, [14](#)