

# Package ‘GREENeR’

September 7, 2022

**Type** Package

**Title** Geospatial Regression Equation for European Nutrient Losses (GREEN)

**Version** 0.1.1

**Author** A. Udias [aut],  
B. Grizzetti [aut],  
O. Vigiak [aut],  
J. Gomez [aut],  
C. Alfaro [aut, cre],  
A. Aloe [aut]

**Maintainer** C. Alfaro <c.alfarog@gmail.com>

**Description** Tools and methods to apply the model Geospatial Regression Equation for European Nutrient losses (GREEN);  
Grizzetti et al. (2005) <[doi:10.1016/j.jhydrol.2004.07.036](https://doi.org/10.1016/j.jhydrol.2004.07.036)>;  
Grizzetti et al. (2008);  
Grizzetti et al. (2012) <[doi:10.1111/j.1365-2486.2011.02576.x](https://doi.org/10.1111/j.1365-2486.2011.02576.x)>;  
Grizzetti et al. (2021) <[doi:10.1016/j.gloenvcha.2021.102281](https://doi.org/10.1016/j.gloenvcha.2021.102281)>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** hydroGOF (>= 0.4-0), FME (>= 1.3.6.1), data.table (>= 1.13.6),  
reshape2 (>= 1.4.4), ggplot2 (>= 3.3.5), graphics (>= 3.6.1),  
sf (>= 1.0-2), dplyr (>= 1.0.7), magrittr (>= 2.0.1), tmap (>= 3.3-2),  
gridExtra (>= 2.3), tidyselect (>= 1.1.0), classInt (>= 0.4-3),  
grDevices (>= 3.5), networkD3 (>= 0.4), parallely (>= 1.30.0)

**Depends** R (>= 3.5.0)

**RoxygenNote** 7.2.1

**URL** <https://github.com/calfarog/GREENeR>

**BugReports** <https://github.com/calfarog/GREENeR/issues>

**Suggests** testthat, knitr, rmarkdown, codetools

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-09-07 12:10:02 UTC

## R topics documented:

GREENeR-package . . . . .	2
annual_data_TN . . . . .	3
annual_data_TP . . . . .	4
calib_boxplot . . . . .	4
calib_dot . . . . .	5
calib_green . . . . .	6
catch_data_TN . . . . .	7
catch_data_TP . . . . .	8
compare_calib . . . . .	8
green_shares . . . . .	10
input_maps . . . . .	11
input_plot . . . . .	12
input_Tserie . . . . .	13
N4_sankey . . . . .	14
nutrient_maps . . . . .	14
nutrient_tserie . . . . .	16
read_geometry . . . . .	17
read_NSdata . . . . .	17
region_nut_balance . . . . .	18
scatter_plot . . . . .	19
select_params . . . . .	20
shreve . . . . .	21
simobs_annual_plot . . . . .	21
<b>Index</b>	<b>23</b>

---

GREENeR-package	<i>GREENeR: Geospatial Regression Equation for European Nutrient Losses</i>
-----------------	---

---

## Description

The package provides tools and methods to apply the model Geospatial Regression Equation for European Nutrient losses (GREEN; Grizzetti et al. (2005); Grizzetti et al. (2012); Grizzetti et al. (2021)) to an area of interest in R environment. The package comprises functions for assessing annual nutrient (nitrogen and phosphorus) loads from a basin or region of interest, land and river retention, and contribution shares by sources. A brief description of the model, including sources and parameters, can be found at the end of this document. Further, the package includes functions for loading spatio-temporal data , calibrating basin parameters, performing an advanced sensitivity

analysis to evaluate the calibration results, and visualizing model inputs and outputs through plots and maps. The package is parallel-capable to alleviate the computational burden in large basins.

## References

- Grizzetti, B., Bouraoui, F., De Marsily, G., & Bidoglio, G. (2005). A statistical method for source apportionment of riverine nitrogen loads. *Journal of Hydrology*, 304(1-4), 302-315. doi: [10.1016/j.jhydrol.2004.07.036](https://doi.org/10.1016/j.jhydrol.2004.07.036)
- Grizzetti, B., Bouraoui, F., De Marsily, G., (2008). Assessing nitrogen pressures on European surface water. *Global Biogeochem. Cycles* 22..
- Grizzetti, B., Bouraoui, F., & Aloe, A. (2012). Changes of nitrogen and phosphorus loads to European seas. *Global Change Biology*, 18(2), 769-782. doi: [10.1111/j.13652486.2011.02576.x](https://doi.org/10.1111/j.13652486.2011.02576.x)
- Grizzetti, B., Vigiak, O., Udias, A., Aloe, A., Zanni, M., Bouraoui, F., Pistocchi, A., Dorati, C., Friedland, R., De Roo, A., others & Bielza, M. (2021). How EU policies could reduce nutrient pollution in European inland and coastal waters. *Global Environmental Change*, 69, 102281. doi: [10.1016/j.gloenvcha.2021.102281](https://doi.org/10.1016/j.gloenvcha.2021.102281)

---

annual\_data\_TN

*Annual data TN*

---

## Description

Defines the sources of nutrient (nitrogen) for each year and catchments.

## Usage

annual\_data\_TN

## Format

A data frame with 14 variables:

BasinID integer. The basin unique identifier.

YearValue integer. The year for which data are defined.

HydroID integer positive. Unique catchment identifier.

NextDownID integer. Unique identifier of the catchment to which the catchment goes.

Atm double. Annual nitrogen deposition from atmosphere (ton/yr).

Min double. Annual amount of nitrogen from mineral fertilisers (ton/yr).

Man double. Annual amount of nitrogen in manure fertilisers (ton/yr).

Fix double. Annual amount of nitrogen fixation by leguminous crops and fodder (ton/yr).

Soil double. Annual amount of nitrogen fixation by bacteria in soils (ton/yr).

Sd double. Nitrogen input from scattered dwellings (ton/yr).

Ps double. Nitrogen input from point sources (ton/yr).

YearlyMass double. Observed annual total nitrogen load (TN ton/yr) from monitoring station data.

ForestFraction double. Non-agricultural land cover in the catchment (fraction).

InvNrmRain double. Inverse of normalized rainfall.

---

annual_data_TP	<i>Annual data TP</i>
----------------	-----------------------

---

### Description

Defines the sources of nutrient (phosphorus) for each year and catchments.

### Usage

```
annual_data_TP
```

### Format

A data frame with 12 variables:

BasinID integer. The basin unique identifier.

YearValue integer. The year for which data are defined.

HydroID integer positive. Unique catchment identifier.

NextDownID integer. Unique identifier of the catchment to which the catchment goes.

Bg double. Annual amount of phosphorus background losses (ton/yr).

Min double. Annual amount of phosphorus mineral fertilisers (ton/yr).

Man double. Annual amount of phosphorus in manure fertilisers (ton/yr).

Sd double. Phosphorus input from scattered dwellings (ton/yr).

Ps double. Phosphorus input from point sources (ton/yr).

YearlyMass double. Observed annual total phosphorus load (TP ton/yr) from monitoring station data.

ForestFraction double. Non-agricultural land cover in the catchment (fraction).

InvNrmRain double. Inverse of normalized rainfall.

---

calib_boxplot	<i>Boxplot of best parameters</i>
---------------	-----------------------------------

---

### Description

Returns boxplots of best model parameters ranked according to different goodness-of-fit measures, and also boxplot with the distribution of the parameters values.

### Usage

```
calib_boxplot(df_cb, rate_bs)
```

**Arguments**

df\_cb            data frame. Table with the result of the calibration process.

rate\_bs        numeric. Rate (%) of parameters selected from the whole set produced in the calibration.

**Value**

Multiple boxplots

**Examples**

```
# the data of the TN scenario
data(catch_data_TP)
data(annual_data_TP)
# the parameter for the calibration of the model
n_iter <- 2 # number of iterations
# the lower limits for all params (alpha_P, alpha_L, sd_coef)
low <- c(10, 0.000, 0.1)
# the upper limits for all params (alpha_P, alpha_L, sd_coef)
upp <- c(70, 0.3, 0.9)
# years in which the model should be executed
years <- 1990:2018
# execution of the calibration
df_calib <- calib_green(catch_data_TP, annual_data_TP, n_iter, low, upp,
                        years)
# Generating the box plots
rateBS <- 5 # rate of best set of parameter to include in the plots
calib_boxplot(df_calib, rateBS)
```

---

calib\_dot

*Dot plot of goodness-of-fit metric vs parameter value*


---

**Description**

Dot plot of goodness-of-fit metric vs parameters value

**Usage**

```
calib_dot(df_cb, param)
```

**Arguments**

df\_cb            data frame. A table with the result of the calibration process.

param           character. Goodness of fit measures. See alternatives link "NSE" "rNSE", "NSE", "mNSE", "MAE", "PBIAS", "cp", "R2".

**Value**

Multiple dot plots

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
# the parameter for the calibration of the model
n_iter <- 2 # number of iterations
# the lower limits for all params (alpha_P, alpha_L, sd_coef)
low <- c(10, 0.000, 0.1)
# the upper limits for all params (alpha_P, alpha_L, sd_coef)
upp <- c(70, 0.3, 0.9)
# years in which the model should be executed
years <- 1990:2018
# execution of the calibration
df_calib <- calib_green(catch_data_TN, annual_data_TN, n_iter, low, upp,
years)
# Generating the dot plots
gof_mes <- "NSE"
calib_dot(df_calib, gof_mes)
```

---

calib\_green

*Calibration of the GREEN model*

---

**Description**

Runs GREEN model calibration

**Usage**

```
calib_green(catch_data, annual_data, n_iter, low, upp, years)
```

**Arguments**

catch_data	data frame. Definition of the topological sequence of catchments.
annual_data	data frame. Sources of nutrient for each year and catchments.
n_iter	numeric. Number of iterations for the calibration process.
low	numeric. Lower bounds of the calibration parameters.
upp	numeric. Upper bounds of the calibration parameters.
years	integer. Years to be used in the calibration. For sequences use c(yearini:yearend).

**Value**

One object, a data frame with the model calibration

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
# the parameter for the calibration of the model
n_iter <- 2 # number of iterations
# the lower limits for all params (alpha_P, alpha_L, sd_coef)
low <- c(10, 0.000, 0.1)
# the upper limits for all params (alpha_P, alpha_L, sd_coef)
upp <- c(70, 0.3, 0.9)
# years in which the model should be executed
years <- 1990:2018
# execution of the calibration
dF_calib <- calib_green(catch_data_TN, annual_data_TN, n_iter, low, upp,
years)
```

---

catch\_data\_TN

*Catch data TN*


---

**Description**

Defines the topological sequence of catchments for nitrogen.

**Usage**

```
catch_data_TN
```

**Format**

A data frame with 5 variables:

HydroID integer positive. Unique catchment identifier.

To\_catch integer. Unique identifier of the catchment to which the catchment goes. Note that for the outlet To\_catch== -1.

Shreve integer. this indicates the Shreve order of the topological sequence in the stream network.

LakeFrRet fraction, 0-1. Lake retention fraction.

NrmLengthKm double. Normalized length of catchment reach.

---

catch_data_TP	<i>Catch data TP</i>
---------------	----------------------

---

**Description**

Defines the topological sequence of catchments for phosphorus.

**Usage**

```
catch_data_TP
```

**Format**

A data frame with 5 variables:

HydroID integer positive. Unique catchment identifier.

To\_catch integer. Unique identifier of the catchment to which the catchment goes. Note that for the outlet To\_catch== -1.

Shreve integer. this indicates the Shreve order of the topological sequence in the stream network.

LakeFrRet fraction, 0-1. Lake retention fraction.

NrmLengthKm double. Normalized length of catchment reach.

---

compare_calib	<i>Plot comparing observed vs modeled loads for two set of parameters</i>
---------------	---

---

**Description**

Returns a scatter plot comparing observed versus modeled loads obtained with two model parameter sets

**Usage**

```
compare_calib(
  catch_data,
  annual_data,
  alpha_p1,
  alpha_l1,
  sd_coef1,
  alpha_p2,
  alpha_l2,
  sd_coef2,
  years,
  name_basin,
  setPlabels
)
```



**Arguments**

catch_data	data frame. Definition of the topological sequence of catchments.
annual_data	data frame. Sources of nutrient for each year and catchments.
alpha_p1	numeric. The basin retention coefficient of the first set of parameters.
alpha_l1	numeric. The river retention coefficient of the first set of parameters.
sd_coef1	numeric. Fraction of domestic diffuse sources that reaches the stream network of the first set of parameters.
alpha_p2	numeric. The basin retention coefficient of the second set of parameters.
alpha_l2	numeric. The river retention coefficient of the second set of parameters.
sd_coef2	numeric. Fraction of domestic diffuse sources that reaches the stream network of the second set of parameters.
years	numeric. Years to be shown in the plot.
name_basin	character. Name of the basin (title of the plot).
setPlabels	character. Labels identifying each set of parameter.

**Value**

A scatter plot and a list with two data frames with model GREEN applied to two model parameter sets

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)

# the first set of parameters to assess the basin model
alpha_p <- 35.09
alpha_l <- 0.02
sd_coef <- 0.2

# the second set of parameters to assess the basin model
alpha_p2 <- 41.23
alpha_l2 <- 0.0015
sd_coef2 <- 0.6

# years in which the plot will we shown
years <- 1990:2018

nameBasin <- "Lay"

# generating the scatter plot comparing two set of parameters observed
# versus modeled loads by year
setPlabels <- c("bestNSE", "bestR2")
compare_calib(catch_data_TN, annual_data_TN, alpha_p , alpha_l, sd_coef,
alpha_p2, alpha_l2, sd_coef2, years, nameBasin, setPlabels)
```

---

green_shares	<i>Geospatial Regression Equation parallel execution returning the source apportionment</i>
--------------	---

---

### Description

Run GREEN model with selected parameter set and returns the nutrient load by each source for all catchments in the Basin.

### Usage

```
green_shares(catch_data, annual_data, alpha_p, alpha_l, sd_coef, loc_years)
```

### Arguments

catch_data	data frame. Definition of the topological sequence of catchments.
annual_data	data frame. Sources of nutrient for each year and catchments.
alpha_p	numeric. First model parameter, the basin retention coefficient.
alpha_l	numeric. Second model parameter, the river retention coefficient.
sd_coef	numeric. Third model parameter, fraction of domestic diffuse sources that reaches the stream network.
loc_years	integer. Years in which the model should be executed.

### Value

One object, a data frame with the nutrient load by each source for all catchments in the Basin

### Examples

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
# the parameter to assess the basin model
alpha_p <- 35.09
alpha_l <- 0.02
sd_coef <- 0.2
# year in which the model should be executed
loc_years <- 1990:2018
# Computing the source apportionment
basin_loads_s <- green_shares(catch_data_TN, annual_data_TN, alpha_p, alpha_l,
sd_coef, loc_years)
```

---

input_maps	<i>Map average load input by source</i>
------------	---

---

**Description**

Map showing the mean load input by source

**Usage**

```
input_maps(
  catch_data,
  annual_data,
  sh_file,
  basin_name,
  plot.type,
  style_map = "fisher",
  scale_barTextS = 0.7,
  legend_position = 1
)
```

**Arguments**

catch_data	data frame. Definition of the topological sequence of catchments.
annual_data	data frame. Sources of nutrient for each year and catchments.
sh_file	sf object. The spatial information.
basin_name	character. The title of the map
plot.type	character. Alternatives of the map: input load (kt) by type divided by year and catchment. "gr1": by km2; "gr2": by year/km2.
style_map	character. Alternatives to create the intervals in the maps. Chosen style: one of "fixed", "sd", "equal", "pretty", "quantile", "kmeans", "hclust", "bclust", "fisher", "jenks".
scale_barTextS	numeric. To modify the size of the text in the legend.
legend_position	numeric. Legend position: 1 (default): "right", "bottom"; 2: "left", "up"; 3: "right", "bottom"; 4: "right", "up".

**Value**

No return value, called for the side effect of drawing a plot

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
```

```

data(sh_file)
# The title of the plot
mapTitle <- "Time series for the Lay Basin"
# the Input Load Map by source type 1 (lines)
input_maps(catch_data_TN, annual_data_TN, sh_file, mapTitle, "gr1",
legend_position = 2)
# the Input Load Map by source type 2 (lines & area)
input_maps(catch_data_TN, annual_data_TN, sh_file, mapTitle, "gr2",
legend_position = 2)

```

---

input\_plot

*Plot input load by source*


---

### Description

A grouped barplot representing the average input load by source for the whole basin or a three density plots showing the distribution of nutrient sources (7 for nitrogen, 5 for phosphorous).

### Usage

```
input_plot(annual_data, sh_file, basin_name, plot.type, coef_SD = 1)
```

### Arguments

annual_data	data frame. Sources of nutrient for each year and catchments.
sh_file	sf object. The spatial information.
basin_name	character. The title of the plot.
plot.type	character. Possible values: Bar plot ("B") or Density plot ("D").
coef_SD	numeric. The standard deviation coefficient.

### Value

No return value, called for the side effect of drawing a plot

### Examples

```

# the data of the TN scenario
data(annual_data_TN)
data(sh_file)
# The name of the basin
basin_name <- "Lay"
# the barplot
input_plot(annual_data_TN, sh_file, basin_name, "B")
# the density plots
input_plot(annual_data_TN, sh_file, basin_name, "D")

```

---

input_Tserie	<i>Time series of annual load inputs by source</i>
--------------	--

---

**Description**

Creates a time series plot showing basin inputs by source

**Usage**

```
input_Tserie(catch_data, annual_data, sh_file, basin_name, plot.type)
```

**Arguments**

catch_data	data frame. Definition of the topological sequence of catchments.
annual_data	data frame. Sources of nutrient for each year and catchments.
sh_file	sf object. The spatial information.
basin_name	character. The title of the plot
plot.type	character. Alternative of the plot: "gr1": stacked area; "gr2": lines & area; "gr3": by km2; "gr4" by km2 and Shreve.

**Value**

A time-series plot

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
data(sh_file)
# The title of the plot
plotTitle <- "Time series for the Lay Basin"
# the time serie plot 1 (lines)
input_Tserie(catch_data_TN, annual_data_TN, sh_file, plotTitle, "gr1")
# the time serie plot 2 (lines & area)
input_Tserie(catch_data_TN, annual_data_TN, sh_file, plotTitle, "gr2")
# the time serie plot 3 (by km2)
input_Tserie(catch_data_TN, annual_data_TN, sh_file, plotTitle, "gr3")
# the time serie plot 4 (by km2 and Shreve)
input_Tserie(catch_data_TN, annual_data_TN, sh_file, plotTitle, "gr4")
```

---

`N4_sankey`*Nutrient balance flow plot*

---

**Description**

Nutrient balance flow in Sankey plot

**Usage**

```
N4_sankey(Nbalance_out)
```

**Arguments**

`Nbalance_out` data frame. Nutrient balance result from the `Nutbalance()` function

**Value**

A Sankey diagram and a data frame with the some variable values

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
# the parameter to assess the basin model
alpha_p <- 35.09
alpha_l <- 0.02
sd_coef <- 0.2
# years in which the model should be executed
loc_years <- 1990:2018
# Computing the nutrient balance
nut_bal <- region_nut_balance(catch_data_TN, annual_data_TN, alpha_p, alpha_l,
sd_coef, loc_years)
# Plot the sankey plot with the result of the balance
sank <- N4_sankey(nut_bal)
```

---

`nutrient_maps`*Map average load output by source*

---

**Description**

Creates maps showing basin output total or by source loads

**Usage**

```
nutrient_maps(green_file, sh_file, basin_name, plot.type, legend_position = 1)
```

**Arguments**

green_file	data frame of GREEN model results from green_shares() function. Nutrient Load by source apportionment of nutrient for each year and catchments.
sh_file	sf object. The spatial information of the basin.
basin_name	character. The title of the map.
plot.type	character. Alternatives of the map: "gr1": output load (kt/y) by source; "gr2": Total Load, log10 (kt/y); "gr3": Total Load by km2 (kt/year/km2).
legend_position	numeric. Legend position: 1 (default): "right", "bottom"; 2: "left", "up"; 3: "right", "bottom"; 4: "right", "up".

**Value**

No return value, called for the side effect of drawing a plot

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
data(sh_file)
# the parameter to assess the basin model
alpha_p <- 35.09
alpha_l <- 0.02
sd_coef <- 0.2
# years in which the model should be executed
loc_years <- 1990:2018
# Computing the source apportionment
basin_sa <- green_shares(catch_data_TN, annual_data_TN, alpha_p, alpha_l,
sd_coef, loc_years)
# The title of the Map
mapTitle <- "Output Loads for the Lay Basin"
# Basin Output Load Maps by source
Lpos <- 1
nutrient_maps(basin_sa, sh_file, mapTitle, "gr1", Lpos)
# Basin Output Specific Load Maps
Lpos <- 1
nutrient_maps(basin_sa, sh_file, mapTitle, "gr2", Lpos)
# Basin Output Specific Load by km2 Maps
Lpos <- 1
nutrient_maps(basin_sa, sh_file, mapTitle, "gr3", Lpos)
```

---

nutrient_tserie	<i>Output load time series plot</i>
-----------------	-------------------------------------

---

### Description

Creates a time series plot showing basin model results

### Usage

```
nutrient_tserie(green_file, sh_file, basin_name, plot.type)
```

### Arguments

green_file	data frame. Nutrient Load by source apportionment of nutrient for each year and catchments.
sh_file	sf object. The spatial information.
basin_name	character. The title of the plot.
plot.type	character. Alternative of the plot: output load (t) by source; gr1: Basin average by Shreve (t/y/km2); gr2: Outlet total (kt/y); gr3: Outlet by source apportionment (kt/y).

### Value

No return value, called for the side effect of drawing a plot

### Examples

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
data(sh_file)
# the parameter to assess the basin model
alpha_p <- 35.09
alpha_l <- 0.02
sd_coef <- 0.2
# years in which the model should be executed
loc_years <- 1990:2018
# Computing the source apportionment
basin_sa <- green_shares(catch_data_TN, annual_data_TN, alpha_p, alpha_l,
sd_coef, loc_years)
# The title of the plot
plotTitle <- "Time series Load Output for the Lay Basin"
# Output Load Basin average time series (lines)
nutrient_tserie(basin_sa, sh_file, plotTitle, "gr1")
# Total Load in the Basin Outlet time series (lines)
nutrient_tserie(basin_sa, sh_file, plotTitle, "gr2")
# Total Load in the Basin Outlet by source apportionment time series (lines)
```



```
nutrient_tserie(basin_sa, sh_file, plotTitle, "gr3")
```

---

read_geometry	<i>Read geometry</i>
---------------	----------------------

---

### Description

Function to read the geometry file.

### Usage

```
read_geometry(file)
```

### Arguments

`file`                    string. A string with the name and extension of the geometry file.

### Value

One object, a sf file.

---

read_NSdata	<i>Read NS data</i>
-------------	---------------------

---

### Description

Function to read the data and return the data frame for GREEN execution.

### Usage

```
read_NSdata(path, tsn, obs, ff, rain, topo, lr, length)
```

### Arguments

`path`                    string. A string with the path of the CSV files.

`tsn`                    file. A CSV file with nine variables YearValue (integer), HydroID (integer), Atm (float), Min (float), Man (float), Fix (float), Soil (float), Sd (float) and Ps (float).

`obs`                    file. A CSV file with three variables YearValue (integer), HydroID (integer) and YearlyMass (float).

`ff`                    file. A CSV file with three variables YearValue (integer), HydroID (integer) and ForestFraction (float).

`rain`                    file. A CSV file with three variables YearValue (integer), HydroID (integer) and Rain (float).

topo	file. A CSV file with two variables HydroID (integer) and Next_HydroID (integer).
lr	file. A CSV file with three variables HydroID (integer), AvgDepth (float) and ResTime (float).
length	file. A CSV file with two variables HydroID (integer) and LengthKm (float).

**Value**

One object, a list with two data frame. First position of the list contains the catch data and the second one the annual data.

**Examples**

```
path <- "https://raw.githubusercontent.com/califarog/GREENeR_data/main/data/csv/"
ns_data <- read_NSdata(path, "TS_nutrients.csv", "Obs_monitoring.csv",
"ForestFr.csv", "Precipitation.csv", "Topology.csv", "LakeProperties.csv",
"Length.csv")
```

---

region_nut_balance	<i>Nutrient balance based in the application of the Geospatial Regression Equation returning the diffuse, land retention, point sources</i>
--------------------	---

---

**Description**

Computes the basin nutrient balance.

**Usage**

```
region_nut_balance(
  catch_data,
  annual_data,
  alpha_p,
  alpha_l,
  sd_coef,
  loc_years,
  atm_coeff = 0.38
)
```

**Arguments**

catch_data	data frame. Definition of the topological sequence of catchments.
annual_data	data frame. Sources of nutrient for each year and catchments.
alpha_p	numeric. First model parameter, the basin retention coefficient.
alpha_l	numeric. Second model parameter, the river retention coefficient.

sd_coef	numeric. Third model parameter, fraction of domestic diffuse sources that reaches the stream network.
loc_years	integer. Years in which the model should be executed.
atm_coeff	numeric. A value for atmospheric attenuation coefficient.

**Value**

One object, a data frame with the basin nutrient balance

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
# the parameter to assess the basin model
alpha_p <- 35.09
alpha_l <- 0.02
sd_coef <- 0.2
# year in which the model should be executed
loc_years <- 1990:2018
# Computing the nutrient balance
basin_loads_b <- region_nut_balance(catch_data_TN, annual_data_TN, alpha_p, alpha_l,
sd_coef, loc_years)
```

---

scatter\_plot

*Scatter plot of goodness-of-fit metric vs parameters*


---

**Description**

Scatter plot of goodness-of-fit metric vs parameters

**Usage**

```
scatter_plot(df_cb, param)
```

**Arguments**

df_cb	data frame. A table with the result of the calibration process.
param	character. Goodness of fit metric: "NSE", "rNSE", "NSE", "mNSE", "MAE", "PBIAS", "cp", "R2",...

**Value**

Multiple scatter plot

**Examples**

```

# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
# the parameter for the calibration of the model
n_iter <- 2 # number of iterations
# the lower limits for all params (alpha_P, alpha_L, sd_coef)
low <- c(10, 0.000, 0.1)
# the upper limits for all params (alpha_P, alpha_L, sd_coef)
upp <- c(70, 0.3, 0.9)
# years in which the model should be executed
years <- 1990:2018
# execution of the calibration
df_calib <- calib_green(catch_data_TN, annual_data_TN, n_iter, low, upp,
years)
gof_mes <- "NSE"
scatter_plot(df_calib, gof_mes)

```

---

select\_params

*Selection of best calibration parameters*


---

**Description**

Return the best calibration parameter set according to one goodness-of-fit metric

**Usage**

```
select_params(df_cb, param)
```

**Arguments**

df\_cb                    data frame. The result of the calibration process.

param                    numeric. Goodness-of-fit measures. "NSE", "rNSE", "NSE", "mNSE", "MAE", "PBIAS", "cp", "R2",...

**Value**

A vector with the 3 parameters

**Examples**

```

# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
# the parameter for the calibration of the model

```

```

n_iter <- 2 # number of iterations
# the lower limits for all params (alpha_P, alpha_L, sd_coef)
low <- c(10, 0.000, 0.1)
# the upper limits for all params (alpha_P, alpha_L, sd_coef)
upp <- c(70, 0.3, 0.9)
# years in which the model should be executed
years <- 1990:2018
# execution of the calibration
df_calib <- calib_green(catch_data_TN, annual_data_TN, n_iter, low, upp,
years)
# Extract the best set of parameter according to a Goodnes of fit metric
gof_mes <- "NSE"
NSE_bestParams <- select_params(df_calib, gof_mes)

```

---

shreve

*Shreve*


---

### Description

Function to read the data and return the data frame for GREEN execution.

### Usage

```
shreve(the_SC)
```

### Arguments

the\_SC            table. A table with topology data.

### Value

One object, a data frame with the shreve.

---

simobs\_annual\_plot

*Scatter plot comparing observed vs modeled loads by year*


---

### Description

Plot

**Usage**

```
simobs_annual_plot(
  catch_data,
  annual_data,
  alpha_p,
  alpha_l,
  sd_coef,
  years,
  name_basin,
  max_value
)
```

**Arguments**

catch_data	data frame. Definition of the topological sequence of catchments.
annual_data	data frame. Sources of nutrient for each year and catchments.
alpha_p	numeric. First model parameter, the basin retention coefficient.
alpha_l	numeric. Second model parameter, the river retention coefficient.
sd_coef	numeric. Third model parameter, fraction of domestic diffuse sources that reaches the stream network.
years	numeric. Years to be shown in the plot.
name_basin	character. The title of the plot.
max_value	numeric. The maximum value for x and y axis.

**Value**

Multiple scatter plot and a data frame with annual nutrient (nitrogen or phosphorus) load for all catchments in the basin

**Examples**

```
# the data of the TN scenario
data(catch_data_TN)
data(annual_data_TN)
# the parameter to assess the basin model
alpha_p <- 35.09
alpha_l <- 0.02
sd_coef <- 0.2
# years in which the plot will we shown
years <- 1990:2018
maxvalue <- 10
# generating the scatter plot comparing observed vs modeled loads by year
name_basin <- "Lay NSE"
max_value <- 10
simobs_annual_plot(catch_data_TN, annual_data_TN, alpha_p, alpha_l,
sd_coef, years, name_basin, max_value)
```

# Index

## \* datasets

- annual\_data\_TN, [3](#)
- annual\_data\_TP, [4](#)
- catch\_data\_TN, [7](#)
- catch\_data\_TP, [8](#)

- annual\_data\_TN, [3](#)
- annual\_data\_TP, [4](#)

- calib\_boxplot, [4](#)
- calib\_dot, [5](#)
- calib\_green, [6](#)
- catch\_data\_TN, [7](#)
- catch\_data\_TP, [8](#)
- compare\_calib, [8](#)

- green\_shares, [10](#)
- GREENeR (GREENeR-package), [2](#)
- GREENeR-package, [2](#)

- input\_maps, [11](#)
- input\_plot, [12](#)
- input\_Tserie, [13](#)

- N4\_sankey, [14](#)
- nutrient\_maps, [14](#)
- nutrient\_tserie, [16](#)

- read\_geometry, [17](#)
- read\_NSdata, [17](#)
- region\_nut\_balance, [18](#)

- scatter\_plot, [19](#)
- select\_params, [20](#)
- shreve, [21](#)
- simobs\_annual\_plot, [21](#)