Package ‘GREENeR’

January 27, 2022

Type Package

Title Geospatial Regression Equation for European Nutrient Losses (GREEN)

Version 0.1.0

Author A. Udias [aut],
B. Grizzetti [aut],
O. Viggiak [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. (2008);

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), parallelly (>=
1.30.0)

Depends R (>= 3.5.0)

RoxygenNote 7.1.2

URL https://github.com/calfarog/GREENeR

BugReports https://github.com/calfarog/GREENeR/issues

Suggests testthat, knitr, rmarkdown
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

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**  
*Annual data TP*

**Description**

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

**Usage**

```r
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**  
*Boxplot of best parameters*

**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**

```r
calib_boxplot(df_cb, rate_bs)
```
**calib_dot**

**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**

```r
# 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)
```

---

**Description**

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

**Usage**

```r
calib_dot(df_cb, par)
```

**Arguments**

- `df_cb` data frame. A table with the result of the calibration process.
- `par` character. Goodness of fit measures. See alternatives link "NSE" "rNSE", "NSE", "mNSE", "MAE", "PBIAS", "cp", "R2".
Value

Multiple dot plots

Examples

```r
# 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

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>catch_data</td>
<td>data frame. Definition of the topological sequence of catchments.</td>
</tr>
<tr>
<td>annual_data</td>
<td>data frame. Sources of nutrient for each year and catchments.</td>
</tr>
<tr>
<td>n_iter</td>
<td>numeric. Number of iterations for the calibration process.</td>
</tr>
<tr>
<td>low</td>
<td>numeric. Lower bounds of the calibration parameters.</td>
</tr>
<tr>
<td>upp</td>
<td>numeric. Upper bounds of the calibration parameters.</td>
</tr>
<tr>
<td>years</td>
<td>integer. Years to be used in the calibration. For sequences use c(yearini:yearend).</td>
</tr>
</tbody>
</table>
Value

One object, a data frame with the model calibration

Examples

```r
# 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**

```r
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

Description
Defines the topological sequence of catchments for phosphorus.

Usage

```r
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

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

Usage

```r
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

```r
# 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 be 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

**Description**

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

**Usage**

```r
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**

```r
# 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

Map average load input by source

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.
anual_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)
### 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

```r
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

```r
# 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

Time series of annual load inputs by source

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**

```r
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 same variable values

**Examples**

```r
# 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 <- 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

```r
# 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  

Output load time series plot

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)
nut_balance

nut_balance(basin_sa, sh_file, plotTitle, "gr3")

---

**Description**

Computes the basin nutrient balance.

**Usage**

```r
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**

```r
# 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 <- 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**

```r
# 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)
```
select_params

```
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, par)
```

**Arguments**

- **df_cb**: data frame. The result of the calibration process.
- **par**: numeric. Goodness-of-fit measures. "NSE", "rNSE", "NSE", "mNSE", "MAE", "PBIAS", "cp", "R2"....

**Value**

A vector with the 3 parameters

**Examples**

```r
# 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 Goodness of fit metric
gof_mes <- "NSE"
NSE_bestParams <- select_params(df_calib, gof_mes)
```
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
)

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.

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
# generating the scatter plot comparing observed vs modeled loads by year
name_basin <- "Lay NSE"
simobs_annual_plot(catch_data_TN, annual_data_TN, alpha_p, alpha_l,
                    sd_coef, years, name_basin)
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
nut_balance, 17
nutrient_maps, 14
nutrient_tserie, 16

scatter_plot, 18
select_params, 19
simobs_annual_plot, 20