Package 'OpenLand'

October 12, 2022

Title Quantitative Analysis and Visualization of LUCC

Version 1.0.2

Description Tools for the analysis of land use and cover (LUC) time series. It includes support for loading spatiotemporal raster data and synthesized spatial plotting. Several LUC change (LUCC) metrics in regular or irregular time intervals can be extracted and visualized through one- and multistep sankey and chord diagrams. A complete intensity analysis according to Aldwaik and Pontius (2012) <doi:10.1016/j.landurbplan.2012.02.010> is implemented, including tools for the generation of standardized multilevel output graphics.

License GPL-3

URL https://github.com/reginalexavier/OpenLand

BugReports https://github.com/reginalexavier/OpenLand/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Depends R (>= 3.4.0)

Imports dplyr (>= 0.8.3), tidyr (>= 1.0.0), ggplot2 (>= 3.2.1), gridExtra (>= 2.3), grid, circlize (>= 0.4.8), networkD3 (>= 0.4), raster (>= 3.0.7), methods

Collate 'OpenLand-package.R' 'rasters_input.R' 'demolandscape.R' 'contingencyTable.R' 'data.R' 'generalfunctions.R' 'intensityClass.R' 'generic_method.R' 'intensityAnalysis.R' 'plotMethods.R' 'otherplots.R'

Suggests tmap, knitr, rmarkdown, covr, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Reginal Exavier [aut, cre] (<https://orcid.org/0000-0002-5237-523X>), Peter Zeilhofer [aut]

Maintainer Reginal Exavier <reginalexavier@rocketmail.com>

Repository CRAN

Date/Publication 2021-11-02 07:20:02 UTC

R topics documented:

`OpenLand-package` *OpenLand: land use and cover (LUC) time series analysis in R.*

Description

OpenLand is an open-source R package for the analysis of land use and cover (LUC) time series. It includes support for consistency check and loading spatiotemporal raster data and synthesized spatial plotting. Several LUC change (LUCC) metrics in regular or irregular time intervals can be extracted and visualized through one- and multistep sankey and chord diagrams. A complete intensity analysis according to *(Aldwaik and Pontius, 2012)* is implemented, including tools for the generation of standardized multilevel output graphics.

Author(s)

Reginal Exavier <reginalexavier@rocketmail.com>, Peter Zeilhofer <zeilhoferpeter@gmail.com>

References

Aldwaik, S. Z. and Pontius, R. G. (2012) 'Intensity analysis to unify measurements of size and stationarity of land changes by interval, category, and transition, Landscape and Urban Planning. Elsevier B.V., 106(1), pp. 103–114. doi: 10.1016/j.landurbplan.2012.02.010.

See Also

The core functions in this package: `intensityAnalysis`, `contingencyTable`,

acc_changes *Accumulates changes in a LULC raster time series*

Description

This function calculates the number of times a pixel has changed during the analysed period. It returns a raster with the number of changes as pixel value and a table containing the areal percentage of every pixel value (number of changes).

Usage

```
acc_changes(path)
```

Arguments

path	The path for the Raster* directory or list of Raster* to be analysed.

Value

Two objects, a RasterLayer and a table.

Examples

```
url <- "https://zenodo.org/record/3685230/files/SaoLourencoBasin.rda?download=1"
temp <- tempfile()
download.file(url, temp, mode = "wb") # downloading the SaoLourencoBasin dataset
load(temp)
# the acc_changes() function, with the SaoLourencoBasin dataset
acc_changes(SaoLourencoBasin)
```

barplotLand *Area of LUC categories at time points*

Description

A grouped barplot representing the areas of LUC categories at each time point of the analysed period.

Usage

```
barplotLand(
  dataset,
  legendtable,
  title = NULL,
  caption = "LUC Categories",
  xlab = "Year",
  ylab = "Area (km2 or pixel)",
  area_km2 = TRUE,
  ...
)
```

Arguments

dataset	A table of the multi step transitions (lulc_Multistep) generated by contingencyTable.
legendtable	A table containing the LUC legend items and their respective color (tb_legend).
title	character. The title of the plot.
caption	character. The caption of the plot.
xlab	character. Label for the x axis.
ylab	character. Label for the y axis.
area_km2	logical. If TRUE the change is computed in km2, if FALSE in pixel counts.
...	additional themes parameters, see theme.

Value

a barplot

See Also

ggplot2::theme

Examples

```
# editing the category names

SL_2002_2014$tb_legend$categoryName <- factor(c("Ap", "FF", "SA", "SG", "aa", "SF",
                                                "Agua", "Iu", "Ac", "R", "Im"),
                                      levels = c("FF", "SF", "SA", "SG", "aa", "Ap",
                                                 "Ac", "Im", "Iu", "Agua", "R"))

SL_2002_2014$tb_legend$color <- c("#FFE4B5", "#228B22", "#00FF00", "#CAFF70",
                                  "#EE6363", "#00CD00", "#436EEE", "#FFAEB9",
                                  "#FFA54F", "#68228B", "#636363")
# the plot
barplotLand(dataset = SL_2002_2014$lulc_Multistep,
            legendtable = SL_2002_2014$tb_legend,
            area_km2 = TRUE)
```

`Category-class` *Class Category*

Description

A S4 class for the Category level result of an Intensity analysis. Can be plotted with the plot method `plot`.

Details

The slots `categoryData` and `categoryStationarity` can receive tables for "Gain" or "Loss" in the following format:

1. Gain
   - `categoryData`: `<tibble>`. A table containing 6 columns:
     (a) Period: `<fct>`. The period *[Yt, Yt+1]*.
     (b) To: `<fct>`. A LUC category *j*.
     (c) Interval: `<int>`. Duration of the period *[Yt, Yt+1]* in years.
     (d) GG_km2/GG_pixel: `<dbl>/<int>`. Area of gross gain of category *j* during *[Yt, Yt+1]*.
     (e) Gtj: `<dbl>`. Annual intensity of gross gain of category *j* for time interval *[Yt, Yt+1]*.
     (f) St: `<dbl>`. Annual intensity of change for time interval *[Yt, Yt+1]*.
   - categoryStationarity: `<tibble>`. A table with the results of a stationarity test of the gain of the categories on the Category level, containing 5 columns:
     (a) To: `<fct>`. A category of interest *j*.
     (b) gain: `<int>`. Number of times a category had gains during all time intervals *[Y1, YT]*.
     (c) N: `<int>`. Total number of evaluated time points (T).
     (d) Stationarity: `<chr>`. *Active Gain* or *Dormant Gain*.
     (e) Test: `<chr>`. *Y* if stationarity was detected and *N* if not.
2. Loss
   - `categoryData`: `<tibble>`. A table containing 6 columns:
     (a) Period: `<fct>`. The period *[Yt, Yt+1]*.
     (b) From: `<fct>`. A LUC category *i*.
     (c) Interval: `<int>`. Duration of the period *[Yt, Yt+1]* in years.
     (d) GG_km2/GG_pixel: `<dbl>/<int>`. Area of gross loss of category *i* during *[Yt, Yt+1]*.
     (e) Lti: `<dbl>`. Annual intensity of gross loss of category *i* for time interval *[Yt, Yt+1]*.
     (f) STt: `<dbl>`. Annual intensity of change for time interval *[Yt, Yt+1]*.
   - categoryStationarity: `<tibble>`. A table of stationarity test over the loss of the categories in the Category level, containing 5 columns:

(a) From: <fct>. A category of interest *i*.

(b) loss: <int>. Number of times a category had losses during all time intervals *[Y1, YT]*.

(c) N: <int>. Total number of evaluated time points (T).

(d) Stationarity: <chr>. *Active Loss* or *Dormant Loss*.

(e) Test: <chr>. *Y* if stationarity was detected and *N* if not.

Slots

`lookupcolor` The colors (character vector) associated with the LUC legend items.

`categoryData` tibble. A table of Category level's results (gain (*Gtj*) or loss (*Lti*) values).

`categoryStationarity` tibble. A table containing results of a stationarity test. A change is considered stationary only if the intensities for all time intervals reside on one side of the uniform intensity, i.e are smaller or bigger than the uniform rate over the whole period.

`chordDiagramLand` *One step transitions (Chord diagram)*

Description

A circlize plot representing the one step transitions between two times point of interest.

Usage

```
chordDiagramLand(
  dataset,
  legendtable,
  legposition = c(x = -1.3, y = 0),
  legtitle = "Categories",
  sectorcol = "gray80",
  area_km2 = TRUE,
  legendsize = 1,
  y.intersp = 1,
  x.margin = c(-1, 1)
)
```

Arguments

`dataset`	A table of the one step transition (`lulc_OneStep`) generated by `contingencyTable`.
`legendtable`	A table containing the LUC legend items and their respective color (`tb_legend`).
`legposition`	numeric. A vector containing the 'x' and 'y' values for the position of the legend. (see `legend`).
`legtitle`	character. The title of the legend.
`sectorcol`	character. The color of the external sector containing the years of compared time points.

area_km2	logical. If TRUE the change is computed in km2, if FALSE in pixel counts.
legendsize	numeric. Font size of the legend. (see "cex" in legend).
y.intersp	numeric. character interspacing factor for vertical (y) spacing in the legend.
x.margin	numeric vector ensuring additional space (blank area) on the left or right of the circle for the legend, by default it is c(-1, 1). (see "canvas.xlim" in circos.par)

Value

A Chord Diagram

Examples

```
# editing the category names

SL_2002_2014$tb_legend$categoryName <- factor(c("Ap", "FF", "SA", "SG", "aa", "SF",
                                                "Agua", "Iu", "Ac", "R", "Im"),
                                      levels = c("FF", "SF", "SA", "SG", "aa", "Ap",
                                                 "Ac", "Im", "Iu", "Agua", "R"))

SL_2002_2014$tb_legend$color <- c("#FFE4B5", "#228B22", "#00FF00", "#CAFF70",
                                  "#EE6363", "#00CD00", "#436EEE", "#FFAEB9",
                                  "#FFA54F", "#68228B", "#636363")

# the plot
chordDiagramLand(dataset = SL_2002_2014$lulc_Onestep,
                 legendtable = SL_2002_2014$tb_legend)
```

---

contingencyTable *Contingency table*

Description

Extracts LUC transitions for all input grids of the time series.

Usage

```
contingencyTable(input_raster, pixelresolution = 30)
```

Arguments

input_raster	path (character), Raster* object or list of Raster* objects. See raster for more information about supported file types.
pixelresolution	numeric. The pixel spatial resolution in meter.

Value

A list that contains 5 objects.

- `lulc_Mulstistep`: <tibble> Contingency table for all analysed time steps, containing 8 columns:
  1. Period: <chr> The period *[Yt, Yt+1]*.
  2. From: <dbl> numerical code of a LUC category *i*.
  3. To: <dbl> numerical code of a LUC category *j*.
  4. km2: <dbl> Area in square kilometers that transited from the category *i* to category *j* in the period from *Yt* to *Yt+1*.
  5. Interval: <dbl> Interval of years between the first and the last year of the period *[Yt, Yt+1]*.
  6. QtPixel: <int> Pixel count that transited from the categories *i* to category *j* in the period from *Yt* to *Yt+1*.
  7. yearFrom: <chr> The year that the change comes from *[Yt]*.
  8. yearTo: <chr> The year that the change goes for *[Yt+1]*.
- `lulc_Onestep`:<tibble> Contingency table for the entire analysed period *[Y1, YT]*, containing 8 columns identical with `lulc_Mulstistep`.
- `tb_legend`: <tibble> A table of the pixel value, his name and color containing 3 columns:
  1. categoryValue: <dbl> the pixel value of the LUC category.
  2. categoryName: <factor> randomly created string associated with a given pixel value of a LUC category.
  3. color: <chr> random color associated with the given pixel value of a LUC category. Before further analysis, one would like to change the `categoryName` and `color` values.
     - Therefore the category names have to be in the same order as the `categoryValue` and the `levels` should be put in the right order for legend plotting. Like:

```
myobject$tb_legend$categoryName <- factor(c("name1", "name2", "name3", "name4"),
                                          levels = c("name3", "name2", "name1", "name4"))
```

     - The colors have to in the same order as the values in the `categoryValue` column. Colors can be given by the color name (eg. "black") or an HEX value (eg. #FFFFFF). Like:

```
myobject$tb_legend$color <- c("#CDB79E", "red", "#66CD00", "yellow")
```

- `totalArea`: <tibble> A table with the total area of the study area containing 2 columns:
  1. area_km2: <numeric> The total area in square kilometers.
  2. QtPixel: <numeric> The total area in pixel counts.
- `totalInterval`: <numeric> Total interval of the analysed time series in years.

Examples

```
url <- "https://zenodo.org/record/3685230/files/SaoLourencoBasin.rda?download=1"
temp <- tempfile()
```

```
download.file(url, temp, mode = "wb") #downloading the online dataset
load(temp)
# the contingencyTable() function, with the SaoLourencoBasin dataset
contingencyTable(input_raster = SaoLourencoBasin, pixelresolution = 30)
```

`intensityAnalysis` *Performs the intensity analysis based on cross-tabulation matrices of each time step*

Description

This function implements an Intensity Analysis (IA) according to Aldwaik & Pontius (2012), a quantitative method to analyze time series of land use and cover (LUC) maps. For IA, a cross-tabulation matrix is composed for each LUC transition step in time.

Usage

```
intensityAnalysis(dataset, category_n, category_m, area_km2 = TRUE)
```

Arguments

`dataset`	list. The result object from `contingencyTable`.
`category_n`	character. The gaining category in the transition of interest (n).
`category_m`	character. The losing category in the transition of interest (m).
`area_km2`	logical. If TRUE the change is computed in km2, if FALSE in pixel counts.

Details

IA includes three levels of analysis of LUC changes. Consecutive analysis levels detail hereby information given by the previous analysis level *(Aldwaik and Pontius, 2012, 2013)*.

1. The *interval level* examines how the size and speed of change vary across time intervals.
2. The *category level* examines how the size and intensity of gross losses and gross gains in each category vary across categories for each time interval.
3. The *transition level* examines how the size and intensity of a category's transitions vary across the other categories that are available for that transition.

At each analysis level, the method tests for stationarity of patterns across time intervals.

The function returns a list with 6 objects:

1. lulc_table: `tibble`. Contingency table of LUC transitions at all analysed time steps, containing 6 columns:
   (a) Period: `<fct>`. Evaluated period of transition in the format `year t - year t+1`.

(b) From: <fct>. The category in year t.

(c) To: <fct>. The category in year t+1.

(d) km2: <dbl>. Area in square kilometers that transited from the category From. to the category To in the period.

(e) QtPixel: <int>. Number of pixels that transited from. the category From to the category To in the period.

(f) Interval: <int>. Interval in years of the evaluated period.

2. *lv1_tbl*: An Interval object containing the *St* and *U* values.
3. *category_lvlGain*: A Category object containing the gain of the LUC category in a period (*Gtj*).
4. *category_lvlLoss*: A Category object containing the loss of the LUC category in a period (*Lti*).
5. *transition_lvlGain_n*: A Transition object containing the annualized rate of gain in *category n* (*Rtin*) and the respective Uniform Intensity (*Wtn*).
6. *transition_lvlLoss_m*: A Transition object containing the annualized rate of loss in *category m* (*Qtmj*) and the respective Uniform Intensity (*Vtm*).

Value

Intensity object

References

Aldwaik, S. Z. and Pontius, R. G. (2012) 'Intensity analysis to unify measurements of size and stationarity of land changes by interval, category, and transition, Landscape and Urban Planning. Elsevier B.V., 106(1), pp. 103–114. doi: 10.1016/j.landurbplan.2012.02.010.

Aldwaik, S. Z. and Pontius, R. G. (2013) 'Map errors that could account for deviations from a uniform intensity of land change, International Journal of Geographical Information Science. Taylor & Francis, 27(9), pp. 1717–1739. doi: 10.1080/13658816.2013.787618.

Examples

```
# editing the category name

SL_2002_2014$tb_legend$categoryName <- factor(c("Ap", "FF", "SA", "SG", "aa", "SF",
                                                "Agua", "Iu", "Ac", "R", "Im"),
                                      levels = c("FF", "SF", "SA", "SG", "aa", "Ap",
                                                "Ac", "Im", "Iu", "Agua", "R"))

SL_2002_2014$tb_legend$color <- c("#FFE4B5", "#228B22", "#00FF00", "#CAFF70",
                                  "#EE6363", "#00CD00", "#436EEE", "#FFAEB9",
                                  "#FFA54F", "#68228B", "#636363")

intensityAnalysis(dataset = SL_2002_2014, category_n = "Ap", category_m = "SG", area_km2 = TRUE)
```

Interval-class *Class Interval*

Description

A S4 class for the Interval level result of an Intensity analysis. Can be plotted with the plot method `plot`.

Details

The slot `intervalData` receives a table containing 4 columns in the following format:

1. Period: <fct>. The period of interest *[Yt, Yt+1]*.
2. PercentChange: <dbl>. Changed area on the Interval level (%).
3. St: <dbl>. Annual intensity of change for a time period [Yt, Yt+1].
4. U: <dbl>. Uniform intensity for a LUC category change in a time period of interest.

Slots

`intervalData` tibble. A table with the results of an Intensity analysis at the Interval level (*St* and *U* values).

netgrossplot *Net and gross changes of LUC categories*

Description

A stacked barplot showing net and gross changes of LUC categories during the entire analysed time period.

Usage

```
netgrossplot(
  dataset,
  legendtable,
  title = NULL,
  xlab = "LUC category",
  ylab = "Area (Km2)",
  legend_title = "Changes",
  changesLabel = c(GC = "Gross change", NG = "Net gain", NL = "Net loss"),
  color = c(GC = "gray70", NG = "#006400", NL = "#EE2C2C"),
  area_km2 = TRUE
)
```

Arguments

dataset	A table of the multi step transition (lulc_Mulstistep) generated by contingencyTable.
legendtable	A table containing the LUC legend items and their respective color (tb_legend).
title	character. The title of the plot (optional), use NULL for no title.
xlab	character. Label for the x axis.
ylab	character. Label for the y axis.
legend_title	character. The title of the legend.
changesLabel	character. Labels for the three types of changes, defaults are c(GC = "Gross change", NG = "Net gain", NL = "Net loss").
color	character. A vector defining the three bar colors.
area_km2	logical. If TRUE the change is computed in km2, if FALSE in pixel counts.

Value

A bar plot

Examples

```
# editing the category names

SL_2002_2014$tb_legend$categoryName <- factor(c("Ap", "FF", "SA", "SG", "aa", "SF",
                                                "Agua", "Iu", "Ac", "R", "Im"),
                                      levels = c("FF", "SF", "SA", "SG", "aa", "Ap",
                                                 "Ac", "Im", "Iu", "Agua", "R"))

# the plot
netgrossplot(dataset = SL_2002_2014$lulc_Multistep,
             legendtable = SL_2002_2014$tb_legend,
             title = NULL,
             xlab = "LUC Category",
             changes = c(GC = "Gross changes", NG = "Net Gain", NL = "Net Loss"),
             color = c(GC = "gray70", NG = "#006400", NL = "#EE2C2C"))
```

---

plot *Methods for function* plot *in package* **OpenLand**

Description

Plot Intensity objects based on Intensity Analysis output.

Usage

```
plot(x, y, ...)

## S4 method for signature 'Interval,ANY'
plot(
  x,
  y,
  labels = c(leftlabel = "Interval Change Area (percent of map)", rightlabel =
    "Annual Change Area (percent of map)"),
  title = NA,
  labs = c(type = "Changes", ur = "Uniform Intensity"),
  marginplot = c(lh = -10, rh = 0),
  leg_curv = c(x = 0.1, y = 0.1),
  color_bar = c(fast = "#B22222", slow = "#006400", area = "gray40"),
  fontsize_ui = 10,
  ...
)

## S4 method for signature 'Category,ANY'
plot(
  x,
  y,
  labels = c(leftlabel = "Annual Change Area (km2 or pixels)", rightlabel =
    "Annual Change Intensity (percent of category)"),
  title = NA,
  labs = c(type = "Categories", ur = "Uniform Intensity"),
  marginplot = c(lh = 0.5, rh = 0.5),
  leg_curv = c(x = 0.1, y = 0.1),
  fontsize_ui = 10,
  ...
)

## S4 method for signature 'Transition,ANY'
plot(
  x,
  y,
  labels = c(leftlabel = "Annual Transition Area (km2 or pixels)", rightlabel =
    "Annual Transition Intensity (percent of category)"),
  title = NA,
  labs = c(type = "Categories", ur = "Uniform Intensity"),
  marginplot = c(lh = 0.5, rh = 0.5),
  leg_curv = c(x = 0.1, y = 0.1),
  fontsize_ui = 10,
  ...
)
```

Arguments

x	An intensity object generated by intensityAnalysis.
y	ignored.
...	additional arguments for theme parameters from ggplot2, see theme.
labels	character. Left and right axis titles(caption).
title	character. Main title.
labs	character. The lateral legend.
marginplot	numeric. Adjustment of the origins of left and right part of the plots.
leg_curv	numeric. x and y values that control the arrow size and position pointing to the Uniform Intensity vertical line.
color_bar	character. Colors defined for the fast, slow and area bars (only for an Interval object).
fontsize_ui	numeric. Fontsize of the uniform intensity percent in the plot.
Interval	The class.
Category	The class.
Transition	The class.

Value

An intensity graph

---

sankeyLand *Sankey diagram of LUC transitions (one or multistep)*

Description

A sankey showing the one or multi step LUC transitions during the analysed period.

Usage

```
sankeyLand(dataset, legendtable, iterations = 0)
```

Arguments

dataset	A table of the multi step (lulc_Mulstistep). or one step transitions (lulc_OneStep) generated by contingencyTable.
legendtable	A table containing the LUC legend items and their respective color (tb_legend).
iterations	numeric. Number of iterations in the diagram layout for computation of the depth (y-position) of each node. See sankeyNetwork.

Value

A sankey diagram

See Also

sankeyNetwork

Examples

```
# editing the category names

SL_2002_2014$tb_legend$categoryName <- factor(c("Ap", "FF", "SA", "SG", "aa", "SF",
                                                "Agua", "Iu", "Ac", "R", "Im"),
                                      levels = c("FF", "SF", "SA", "SG", "aa", "Ap",
                                                 "Ac", "Im", "Iu", "Agua", "R"))

SL_2002_2014$tb_legend$color <- c("#FFE4B5", "#228B22", "#00FF00", "#CAFF70",
                                  "#EE6363", "#00CD00", "#436EEE", "#FFAEB9",
                                  "#FFA54F", "#68228B", "#636363")

# onestep sankey
sankeyLand(dataset = SL_2002_2014$lulc_Onestep,
           legendtable = SL_2002_2014$tb_legend)

# multistep sankey
sankeyLand(dataset = SL_2002_2014$lulc_Multistep,
           legendtable = SL_2002_2014$tb_legend)
```

SL_2002_2014 *Tables of land use and cover (LUC) in the São Lourenço River Basin (2002 - 2014)*

Description

A list containing five objects created by the contingencyTable function with SaoLourencoBasin as input (SL_2002_2014 <- contingenceTable(input_raster = SaoLourencoBasin, pixelresolution = 30)).

Usage

```
data(SL_2002_2014)
```

Format

A data list with 5 objects:

lulc_Multistep <tibble> Contingency table for all analysed time steps, containing 8 columns:

1. Period: <chr> The period *[Yt, Yt+1]*.
2. From: <int> numerical code of a LUC category *i*.

3. To: <int> numerical code of a LUC category *j*.
4. km2: <dbl> Area in square kilometers that transited from the category *i* to category *j* in the period from *Yt* to *Yt+1*.
5. QtPixel: <int> Pixel count that transited from the categories *i* to category *j* in the period from *Yt* to *Yt+1*.
6. Interval: <int> Interval of years between the first and the last year of the period *[Yt, Yt+1]*.
7. yearFrom: <int> The year that the change comes from *[Yt]*
8. yearTo: <int> The year that the change goes for *[Yt+1]*

lulc_Onstep <tibble> Contingency table for the entire analysed period *[Yt1, YT]*, containing 8 columns identical with lulc_Mulstistep.

tb_legend <tibble> A table of the pixel value, his name and color containing 3 columns:

1. categoryValue: <int> the pixel value of the LUC category.
2. categoryName: <fct> randomly created string associated with a given pixel value of a LUC category.
3. color: <chr> random color associated with the given pixel value of a LUC category.

totalArea <tibble> A table with the total area of the study area containing 2 columns:

1. area_km2: <dbl> The total area in square kilometers.
2. QtPixel: <int> The total area in pixel counts

.

totalInterval <int> Total interval of the analysed time series in years.

Source

https://www.embrapa.br/pantanal/bacia-do-alto-paraguai

---

summary_dir *Summary of multiple parameters in a raster directory*

Description

Listing major charateristics of raster inputs. Those characteristics are the dimensions, the resolution, the extent, the values (min, max) and the coordinate reference system.

Usage

```
summary_dir(path)
```

Arguments

path	The path for the Raster* directory or list of Raster* to be analysed.

Value

Table with the raster parameters in columns

Examples

```
url <- "https://zenodo.org/record/3685230/files/SaoLourencoBasin.rda?download=1"
temp <- tempfile()
download.file(url, temp, mode = "wb") # downloading the SaoLourencoBasin dataset
load(temp)
# the acc_changes() function, with the SaoLourencoBasin dataset

summary_dir(raster::unstack(SaoLourencoBasin))
```

---

summary_map *Quantitative summary of a unique categorical raster*

Description

This function presents a summary with the pixel quantity of each category present in a categorical raster.

Usage

```
summary_map(path)
```

Arguments

path	The path for the raster to be analysed, if path is a multilayer raster only the first RasterLayer will be analysed.

Value

A table containing in columns the pixel counts for each pixel value

Examples

```
url <- "https://zenodo.org/record/3685230/files/SaoLourencoBasin.rda?download=1"
temp <- tempfile()
download.file(url, temp, mode = "wb") # downloading the SaoLourencoBasin dataset
load(temp)
summary_map(SaoLourencoBasin[[1]])
```

`Transition-class` *Class Transition*

Description

A S4 class for the Transition level result of an Intensity analysis. Can be plotted with the plot method `plot`.

Details

The slots `transitionData` and `transitionStationarity` can receive tables for "Gain of category n" or "Loss of category m" in the following format:

1. Gain of category n:
   - `transitionData`: <tibble>. A table with 7 columns:
     (a) Period: <fct>. The period *[Yt, Yt+1]*.
     (b) From: <fct>. A category i.
     (c) To: <fct>. The gaining category in the transition of interest *(n)*.
     (d) Interval: <int>. Duration of the period *[Yt, Yt+1]*.
     (e) T_i2n_km2/T_i2n_pixel: <dbl>. Area with transition from category i to category n during time interval *[Yt, Yt+1]* where *i* `is not equal to` *n*.
     (f) Rtin: <dbl>. Annual intensity of transition from category i to category n during time interval *[Yt, Yt+1]* where *i* `is not equal to` *n*.
     (g) Wtn: <dbl>. Value of the uniform intensity of the transition to category n from all non-n categories at time Yt during time interval *[Yt, Yt+1]*.
   - transitionStationarity: <tibble>. A table containing results of a stationarity test over the gain on *category n* containing 5 columns:
     (a) From: <fct>. The losing category in the transition of interest to the category n.
     (b) loss: <int>. Number of times the category had losses to the category n.
     (c) N: <int>. Total number of transitions to be considered as stationary (T).
     (d) Stationarity: <chr>. *targeted by* or *avoided by* the category n.
     (e) Test: <chr>. *Y* for stationarity detected and *N* when not.
2. Loss of category m:
   - `transitionData`: <tibble>. A table with 7 columns:
     (a) Period: <fct>. The period *[Yt, Yt+1]*.
     (b) To: <fct>. A category *j*.
     (c) From: <fct>. The losing category in the transition of interest (m).
     (d) Interval: <dbl>. Duration of the period *[Yt, Yt+1]*.
     (e) T_m2j_km2/T_m2j_pixel: <dbl>. Area with transition from category *m* to category *j* during time interval *[Yt, Yt+1]* where *j* `is not equal to` *m*.
     (f) Qtmj: <dbl>. Annual intensity of transition from category *m* to category *j* during time interval *[Yt, Yt+1]* where *j* `is not equal to` *m*.

(g) Vtm: <dbl>. Value of the uniform intensity of the transition from category *m* to all *non-m* categories at time Y_{t+1} during time interval *[Yt, Yt+1]*.

- transitionStationarity: <tibble>. A table containing results of a stationarity test over the loss of *category m* containing 5 columns:
  (a) To: <fct>. The gaining category in the transition of interest from the category m.
  (b) gain: <int>. Number of times the category had gains from the category m.
  (c) N: <int>. Total number of transitions to be considered as stationary (T).
  (d) Stationarity: <chr>. *targeted* or *avoided* the category m.
  (e) Test: <chr>. *Y* for stationarity detected and *N* when not.

Slots

`lookupcolor` The colors (character vector) associated with the LUC legend items.

`transitionData` tibble. A table of Transition level's results (gain n (*Rtin* & *Wtn*) or loss m (*Qtmj* & *Vtm*) values).

`transitionStationarity` tibble. A table containing results of a stationarity test. A change is considered stationary only if the intensities for all time intervals reside on one side of the uniform intensity, i.e are smaller or bigger than the uniform rate over the whole period.

Index