Package 'srn'

November 12, 2018

Title Sub-Regional Nexus Modeling Tool

Version 0.0.1

Description Package to process water-energy-land nexus data to different sub-regional levels.

Depends

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.1.0

Imports
RColorBrewer, rgcam, tibble, dplyr, tmap, ggplot2, scales, utils,tidyr, rlang, grDevices

Remotes github::JGCRI/rgcam

VignetteBuilder knitr

R topics documented:

`srn` *srn: Sub-Regional nexus Package*

Description

The SRN package provides

SRN functions

The SRN functions ...

srn.assumptions *srn.assumptions*

Description

This function loads holds the different assumptions used throughout the srn package.

Usage

```
srn.assumptions()
```

Details

List of Assumptions

- convEJ2TWh
- convEJ2GW
- conv1975USDperGJ22017USDperMWh
- conv1975USDperGJ22017USDperMBTU
- convertGgTgMTC
- GWPType

Value

A list of assumptions

Examples

```
library(srn)
a<-srn.assumptions()
a # will give full list of assumptions
```

srn.chart *srn.chart*

Description

This function produce different kinds of charts for the srn package. iIt requires a table in the SRN format. Each figure is accompanied with a csv table.

Usage

```
srn.chart(srnFormattedTable, chartType = "bar", position = "stack",
  xData = "x", yData = "value", class = "class1",
  group = "scenario", classPalette = "classPalette1",
  classLabel = "classLabel1", xLabel = "xLabel",
  facet_rows = "region", facet_columns = "scenario", ncolrow = 4,
  scales = "fixed", useNewLabels = 1, units = "units",
  xBreaksMaj = 10, xBreaksMin = 5, yBreaksMajn = 5,
  yBreaksMinn = 10, sizeBarLines = 0.5, sizeLines = 1.5)
```

Arguments

`srnFormattedTable`	Table in srn format
`chartType`	Type of chart: "bar" or "line"
`position`	Position in bar charts. "identity", "stack" or "dodge"
`xData`	Default "x"
`yData`	Default "value"
`class`	Default "class1"
`group`	Default "scenario"
`classPalette`	Default "classPalette1"
`classLabel`	Default "classLabel1"
`xLabel`	Default "xLabel"
`facet_rows`	Default "region"
`facet_columns`	Default "scenario"
`ncolrow`	Number of columns or Rows for Faceted plots
`scales`	Default "fixed"
`useNewLabels`	Default 1
`units`	Default "units"
`xBreaksMaj`	Default 10
`xBreaksMin`	Default 5
`yBreaksMajn`	Default 5
`yBreaksMinn`	Default 10
`sizeBarLines`	Default 0.5
`sizeLines`	Default 1.5

Value

Returns the formatted data used to produce chart

`srn.chartsProcess` *srn.chartsProcess*

Description

This function produces charts given any number of tables in the srn format. The srn.chart() function produces charts for each region nd scenario. If there are more than one scenario then the function also produces a folder for diffplots. The input tables should be .csv files with the following columns: scenario, region, sources, param, x, xLabel, vintage, class1, class2, units, value, aggregate, classLabel1,classPalette1,classLabel2,classPalette2. Running the srn.readgcam automatically produces An empty template with these columns for the relevant parameters. Each column is defined below:

Usage

```
srn.chartsProcess(dataTables, scenRef = NULL,
  dirOutputs = paste(getwd(), "/outputs", sep = ""), pdfpng = "png",
  yearsCompare = c("2015", "2030", "2050", "2100"))
```

Arguments

`dataTables`	Vector of strings with full path to datatables to be read in. Example c("D:/srn/outputs/Colombia/regional/dataTable_Colombia_1975to2100.csv", "D:/srn/outputs/Colombia/regional/dataTableLocal_Colombia_1975to2100.csv"). Where "dataTableLocal_Colombia_1975to2100.csv" is the new datafile created based on "dataTableTemplate_Colombia_1975to2100.csv" and contains new local data.
`scenRef`	The reference scenario to compare against. Default will pick first scenario from list f all scenarios
`dirOutputs`	Full path to directory for outputs
`pdfpng`	Choose the format for outputs. Either "pdf", "png" or "both. Default is "png"
`yearsCompare`	Choose the years to compare scenarios for xScenSelectYears plot. Default is c("2015","2030","2050","2100")

Details

List of Assumptions

- scenario: The name of the new data scenario
- region: The region for the data
- sources: Sources for the data
- param: Name of the parameter
- x: The x axis variable values
- xLabel: X axis Label
- vintage: Vintages if any. If not relevant then just enter "Vintage"
- class1: Classes or types (eg. if param is water_demands then the classes may be Industry, Agriculture etc.)
- class2: A second category of classes if exists.
- units: Units for the parameter. These are used as the y axis label.
- value: The parameter value.
- aggregate: Either "sum" or "mean". This paramater is used to determine how to aggregate across regions or scenarios.
- classLabel1: If class1 exists then this will be legend Label. If it doesnt exist enter "classLabel1"
- classPalette1: An R or srn.colors() palette. Can leave the default as "pal_16".
- classLabel2: If class2 exists then this will be legend Label. If it doesnt exist enter "classLabel2"
- classPalette2: An R or srn.colors() palette. Can leave the default as "pal_16".

Value

Produces charts in output folder and also returns combined table in srn format.

srn.colors *srn.colors*

Description

This function loads various color palettes used previously in GCAM as well as new palettes for SRN modeling to the global environment

Usage

```
srn.colors()
```

Details

List of Color Palettes

- pal_16
- elec_tech_colors
- elec_renew_colors
- building_colors
- trn_fuel_colors
- enduse_fuel_numbered
- enduse_colors
- pal_pri_ene
- pal_pri_fuelcost
- pal_emiss_sector
- pal_landuse
- pal_hydrogen
- pal_refliq
- emiss_by_enduse_colors
- biouse_colors
- pal_Basic
- pal_Gas
- pal_Diff
- pal_Diff5
- pal_Absolute
- pal_Absolute5
- pal_Unassigned
- pal_elec_subsec
- pal_elec_finalNrgFuel
- pal_elec_techs
- pal_elec_sec
- pal_finalNrg_sec
- pal_pri_ene
- pal_elec_tech_colors

Value

A list of color palettes.

Examples

```
library(srn)
a<-srn.colors()
pie(rep(1,length(a$pal_Basic)),label=names(a$pal_Basic),col=a$pal_Basic)
```

`srn.readgcam` *srn.readgcam*

Description

This function connects to a gcamdatabase and uses a query file to out results into a table ready for plotting.

Usage

```
srn.readgcam(gcamdatabasePath, gcamdatabaseName,
  queryxml = "srnQueries.xml", scenOrigNames, scenNewNames = NULL,
  reReadData = T, dataProj = "dataProj.proj",
  dirOutputs = paste(getwd(), "/outputs", sep = ""), regions = NULL)
```

Arguments

`gcamdatabasePath`	Path to gcam database folder
`gcamdatabaseName`	Name of gcam database
`queryxml`	Full path to query.xml file
`scenOrigNames`	Original Scenarios names in GCAM database in a string vector. For example c('scenario1','scenario2).
`scenNewNames`	New Names which may be shorter and more useful for figures etc. Default will use Original Names. For example c('scenario1','scenario2)
`reReadData`	If TRUE will read the GCAM data base and create a queryData.proj file in the same folder as the GCAM database. If FALSE will load a '.proj' file if a file with full path is provided otherwise it will search for a dataProj.proj file in the existing folder which may have been created from an old run.
`dataProj`	Optional. A default 'dataProj.proj' is produced if no .Proj file is specified.
`dirOutputs`	Full path to directory for outputs
`regions`	The regions to analyze in a vector. Example c('Colombia','Pakistan')

Value

A list with the scenarios in the gcam database, queries in the queryxml file and a tibble with gcam data formatted for srn charts.

srn.templates *srn.templates*

Description

This script holds various templates used for different scripts.

Usage

```
srn.printPdfPng(figure, dir, filename, figWidth = 13, figHeight = 9,
  pdfpng = "png")

srn.chartsThemeLight()

srn.tmapAnimate(map, filename = "animation.gif", width, height,
  delay = 60)

srn.tmapLayout()
```

Arguments

`figure`	Figure to be printed in function srn.printPdfPng
`dir`	Directory to print figure to in function srn.printPdfPng
`filename`	Filename for figure printed in function srn.printPdfPng
`figWidth`	Figure Width in inches for figures to be printed in function srn.printPdfPng
`figHeight`	Figure height in inches for figures to be printed in function srn.printPdfPng
`pdfpng`	Either "pdf", "png" or "both" to define the format of output
`map`	A tmap object with facets which will be converted to animations
`width`	Width of map in inches.
`height`	Hieght of map
`delay`	Delay. Time between animations = delay/100. Default is 60 or 0.6 seconds.

Details

List of Templates in this script:

- srn.printPdfPng: Function used to print charts to a pdf or png or both.
- srn.chartsThemeLight: A light ggplot theme for charts
- srn.tmapAnimate: A function to animate tmaps across a variable.
- srn.tmapLayout: A fucntion to define tmap layouts

Value

A list of different templates

Index