Package: mredgebuildings (via r-universe)

August 29, 2024

Type Package

Title Prepare data to be used by the EDGE-Buildings model

Version 0.7.0

Date 2024-08-29

Description Prepare data to be used by the EDGE-Buildings model.

License MIT + file LICENSE

URL `https://github.com/pik-piam/mredgebuildings`

Depends madrat (>= 2.8.1), magclass (>= 6.11.0), mrdrivers (>= 1.0.0), mrcommons (>= 1.35.0), mrremind (>= 0.175.4)

Imports brick (>= 0.5.0), dplyr, GDPuc, ncdf4, openxlsx, purrr, plyr, pracma, raster, rlang, stringr, terra, quadprog, quitte, readxl, tidyr, zoo

Suggests covr, lucode2, pdftools, testthat

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://pik-piam.r-universe.dev

RemoteUrl https://github.com/pik-piam/mredgebuildings

RemoteRef HEAD

RemoteSha 61b80bdb838724126d332c96a2749dffc1f2ab19

Contents

mredgebuildings-package *mredgebuildings: Prepare data to be used by the EDGE-Buildings model*

Description

Prepare data to be used by the EDGE-Buildings model.

Author(s)

Maintainer: Robin Hasse `<robin.hasse@pik-potsdam.de>` (ORCID)

Authors:

- Pascal Führlich `<pascal.fuehrlich@pik-potsdam.de>`
- Antoine Levesque
- Hagen Tockhorn

See Also

Useful links:

- `https://github.com/pik-piam/mredgebuildings`

.disaggregate *Disaggregate energy demand within an aggregated region*

Description

Disaggregate regional energy demand per carrier by end use while meeting the end use shares in the aggregated region.

Usage

```
.disaggregate(subset, key)
```

Arguments

`subset`	data frame for one period and aggregated region
`key`	named vector with specification of the subset group (not used)

Details

The function first tries to find a solution that satisfies both the regional carrier and the overall end use constraints. If there is no solution, another optimisation is run that tries to also minimise deviations from the end use quantities but removes them from the constraints.

aggCells *Aggregate cellular HDD/CDD values to country-wide average (population-weighted)*

Description

Aggregate cellular HDD/CDD values to country-wide average (population-weighted)

Usage

```
aggCells(data, weight, mask)
```

Arguments

`data`	raster object containing HDD/CDD values
`weight`	raster object containing aggregation weights
`mask`	raster opbject defining (regional) aggregation boundaries

Value

data frame containing regionally meaned HDD/CDD values

blend *Weighted blend of BAIT and near-surface atmospherical temperature*

Description

To adress loss of buildings' internal memory of previous conditions at high outside temperatures due to window opening, etc., BAIT and outside temperature are blended. The blend is active between a lower and upper temperature threshold, `bLower` and `bUpper`, which are mapped to a range of -5 to +5 of a sigmoid function (corresponding to a 1% and 99% blend). The maximum amount of blending (i.e. the amplitude of the sigmoid function) is given by a parameter `bMax`.

Usage

```
blend(bait, tas, weight)
```

Arguments

`bait`	raster data on BAIT
`tas`	raster data on near-surface atmospherical temperature
`weight`	named list with blending parameters bLower, bUpper, bMax

Value

blended raster data

`calcBAITpars` *calculate regression parameters for BAIT climate variables*

Description

linear regression on historic data to determine regression parameters for surface downdwelling shortwave radiation (rsds), near-surface wind speed (sfcwind) and near-surface specific humidity (huss) with respect to near-surface air temperature (tas). The regression is done with on a simple linear model, where the historical input data covers the years of 2000-2014. For rsds and sfcwind, a simple linear relationship is assumed where for huss an exponential relation is assumed, buildig upon the non-linear relation between water vapor pressure and temperature.

Usage

```
calcBAITpars(model = "GFDL-ESM4")
```

Arguments

`model`	specify GCM responsible for data input

Value

terra SpatRaster covering one regression parameter per layer per cell

Author(s)

Hagen Tockhorn

`calcBuildingStock` *calculate the European historic building stock*

Description

Join data from various sources to construct a consistent European building stock with a focus on floor space (million m2).

Usage

```
calcBuildingStock(subtype = c("residential", "commercial"))
```

Arguments

`subtype`	Character with subsector

Value

MAgPIE object with historic building stock

Note

if the distribution of building types is missing, it is filled with the Europe-wide distribution. It might be worthwhile to map filling countries instead.

Author(s)

Robin Hasse

`calcCarrierPrices` *Calculate carrier prices*

Description

Final energy carrier prices including energy supply cost, transport and distribution cost(T&D) and taxes (without VAT). CO2 price component and VAT are added later depending on the scenario assumptions.

Usage

```
calcCarrierPrices()
```

Author(s)

Robin Hasse

`calcCostConstruction` *calculate construction cost*

Description

floor-space specific construction cost

Usage

```
calcCostConstruction()
```

Author(s)

Robin Hasse

`calcCostDemolition` *Demolition cost*

Description

Demolition cost

Usage

```
calcCostDemolition()
```

Author(s)

Robin Hasse

`calcCostRenovation` *calculate renovation cost*

Description

floor-space specific renovation cost

Usage

```
calcCostRenovation(energyLadder = FALSE)
```

Arguments

`energyLadder`	logical, should the calculation include renovation transitions that are considered a decline on the energy ladder?

Value

MagPIE object with floor-space specific renovation cost depending on the initial and final state of the building

Author(s)

Robin Hasse

`calcDensity` *Get Population Density*

Description

Get Population Density

Usage

```
calcDensity()
```

Value

magpie object

Author(s)

Hagen Tockhorn

`calcEfficiencyRegression` *Calculation Regression Parameters for FE-UE Efficiency Projection*

Description

Calculate the regression parameters for the projection of final (FE) to useful (UE) energy conversion projection for all combinations of energy enduses and carriers. The regression parameters correspond to an asymptotic regression model and encompass the parameters Asym, R0 and lrc. The are determined using a nonlinear least-squares regression.

Usage

```
calcEfficiencyRegression()
```

Details

This approach closely follows the model by De Stercke et al. which is mainly driven by GDP per Capita.

Author(s)

Hagen Tockhorn

References

De Stercke, S. (2014). Dynamics of Energy Systems: A Useful Perspective (Interim Report, p. 68). IIASA. http://pure.iiasa.ac.at/id/eprint/11254/1/IR-14-013.pdf

`calcFEbyEUEC` *Historic Final Energy Demand by Carrier and Enduse*

Description

Takes the historic final energy demand by carriers from IEA and disaggregates it into different end uses.

Usage

```
calcFEbyEUEC()
```

Details

The disaggregation is performed such that the aggregated carrier-specific FE values of the IEA IO regions are met as a minimum requirement. In an ideal case, the enduse-specific FE shares are met as well.

Value

data.frame with historic energy demands

Note

For now, existing disaggregated final energy data with respect to carriers and enduses combined is replaced in the final output. However, since the Odyssee data is largely underestimating the real IEA FE targets, these shall serve as lower boundaries for the disaggregation. Such feature has been implemented in `toolDisaggregate` but is not running smoothly yet.

Author(s)

Hagen Tockhorn, Robin Hasse

`calcFEdemandBuildings` *calculate FE demand in buildings*

Description

Simple estimate of total FE demand in differet building types

Usage

```
calcFEdemandBuildings()
```

Details

Schould only be used as weights not actual demand. The disaggregation to building types is too simplistic.

Author(s)

Robin Hasse

`calcFEUE` *Historical Final and Useful Energy Demand*

Description

Historic Final Energy (FE) Demand and FE-EU-Efficiencies are taken to compute Useful Energy (UE) Demand.

Usage

```
calcFEUE()
```

Value

data.frame with final and useful energy demand

Author(s)

Hagen Tockhorn

`calcFEUEefficiencies` *Calculation and Projection of Final to Useful Energy*

Description

Calculate Efficiencies of Final (FE) to Useful (UE) Energy Conversion for all combinations of Energy Carriers and Enduses. The efficiency projections are based on a model by De Stercke et al. which is mainly driven by GDP per Capita. It describes an S-shaped curve approaching assumed efficiency levels. The parameters of that curve are derived by a regression with observations of IEA data.

Usage

```
calcFEUEefficiencies(gasBioEquality = TRUE)
```

Arguments

`gasBioEquality` Determines if carriers natgas and biomod share the same efficiencies

Author(s)

Hagen Tockhorn

References

De Stercke, S. (2014). Dynamics of Energy Systems: A Useful Perspective (Interim Report, p. 68). IIASA. http://pure.iiasa.ac.at/id/eprint/11254/1/IR-14-013.pdf

`calcFloorspacePast` *Historical residential floor space demand*

Description

Data for many European countries is taken from EEA, India from IEA and other countries from Daioglou et al 2012. The result does not cover all countries and has mixed points in time depending on the region.

Usage

```
calcFloorspacePast()
```

Value

MAgPIE object with historic floor space

Note

RK: In Antoine's EDGE-B, data points associated with an GDP/POP above 70000 USD/cap are dropped here to improve the later regression. This filtering should be moved to getFloorspaceResidential where the regression is performed. Therefore, the high-income data points are kept at this stage.

Author(s)

Robin Hasse, Antoine Levesque, Hagen Tockhorn

`calcFloorspacePerCap` *Residential floor space per capita*

Description

Residential floor space per capita

Usage

```
calcFloorspacePerCap()
```

Value

MagPIE object with residential floor space per capita

Author(s)

Robin Hasse

`calcGDPPop` *Get historic GDP per Capita for SSP Scenarios*

Description

Get historic GDP per Capita for SSP Scenarios

Usage

```
calcGDPPop()
```

Value

magpie object

Author(s)

Hagen Tockhorn

calcHDDCDD *Calculate HDD and CDD based on outdoor/indoor temperature difference*

Description

Heating and cooling degree days based on raw outside temperature or bias-adjusted internal temperature (BAIT), driver for space heating and cooling demand in buildings.

Usage

```
calcHDDCDD(
  mappingFile,
  bait = FALSE,
  multiscen = FALSE,
  rasDir = NULL,
  cacheDir = NULL
)
```

Arguments

`mappingFile`	file name of sectoral mapping containing input data file names and directories
`bait`	boolean, use BAIT instead of raw temperature
`multiscen`	boolean, does `mappingFile` cover more than one scenario?
`rasDir`	absolute path to directory for saving raster files
`cacheDir`	absolute path to directory for pre-calculated BAIT regression parameters

Value

magpie object of heating and cooling degree days

Author(s)

Robin Hasse, Hagen Tockhorn

calcHeatingCapacity *Floor-area specific heating capacity*

Description

Rough assumption based on online reference values scaled with U-values from Hotmaps and heating degree days (HDD).

Usage

```
calcHeatingCapacity(swissFormular = FALSE)
```

Arguments

`swissFormular`	boolean, apply the simplistic swiss formular?

Details

This is a very rough estimation that neglects regional differences and many other effects. It should be replace by bottom-up calculation based on climate data.

Value

MagPIE object with floor-area specific heating capacity

Author(s)

Robin Hasse

Source

https://www.heizung.de/ratgeber/diverses/heizleistung-berechnen-gruende-und-ablauf.html

`calcHeatingSystem` *Calculate installation cost or efficiency of heating systems*

Description

Calculate installation cost or efficiency of heating systems

Usage

```
calcHeatingSystem(subtype = c("Purchasing cost", "Efficiency"))
```

Arguments

`subtype`	character, variable type (either ’Purchasing cost’ or ’Efficiency’)

Value

MagPIE object with capacity-specific purchasing cost or efficiency of heating systems

Author(s)

Robin Hasse

`calcHouseholdSize` *calculate household size*

Description

Number of persons per household.

Usage

```
calcHouseholdSize()
```

Value

MAgPIE object with household size

Note

THE EU Buildings Database gives one share of all dwellings with 5 or more persons. This share is weighted with 5 persons which can cause a downward bias.

Author(s)

Robin Hasse

`calcIEAPFU` *IEA database of primary, final and useful energy*

Description

Computes data for the Primary Final Useful (PFU) resolution

Usage

```
calcIEAPFU()
```

Value

IEA data as MAgPIE object aggregated to country level

Author(s)

Antoine Levesque, Robin Hasse

See Also

`calcOutput`

Examples

```
## Not run:  a <- calcOutput("IEAPFU")
```

---

calcIEA_EEI *Process data from IEA End Uses and Efficiency Indicators Database*

Description

IEA EEI final energy data is processed and mapped w.r.t. carrier and enduse names.

Usage

```
calcIEA_EEI(subtype = c("buildings"))
```

Arguments

`subtype`	sector name

Details

As for the buildings sector, data for residential and commercial ("service") buildings is aggregated and the carrier "biomass" is split into traditional and modern biomass w.r.t. to income per capita.

Value

data.frame containing enduse- and carrier-resoluted energy data

Author(s)

Hagen Tockhorn

---

calcLifetimeParams *Weibull lifetime distribution parameters*

Description

Parameters for the lifetime of heating systems are taken from a very detailed EIA publication for building sector appliances and equipment. The range of the building shell lifetime is taken from Skarning et al. 2017.

Usage

```
calcLifetimeParams(subtype)
```

Arguments

`subtype`	character, type of asset (either `building`, 'heatingSystem' or 'buildingShell')

Value

MagPIE object with Weibull lifetime distribution parameters

Author(s)

Robin Hasse

Source

https://www.eia.gov/analysis/studies/buildings/equipcosts/pdf/full.pdf

http://dx.doi.org/10.1016/j.enbuild.2017.01.080

`calcMatchingReference` *Calculate Reference data for input matching*

Description

Calculate Reference data for input matching

Usage

```
calcMatchingReference(subtype)
```

Arguments

`subtype`	character, matching reference

Author(s)

Robin Hasse

calcPFUDB *Converts Data from Primary Fuel and Useful Energy Data Base (PFUDB)*

Description

PFU Data is disaggregated into different thermal and non-thermal Enduses using FE Shares from Odyssee and IEA_ETP.

Usage

```
calcPFUDB()
```

Details

This was adapted from EDGE function 'getPFUDB.R'.

Value

magpie object

Author(s)

Hagen Tockhorn, Robin Hasse

calcPopulationBuildings *Historic and projected population*

Description

Population differentiated by location (urban/rural) and building type

Usage

```
calcPopulationBuildings()
```

Author(s)

Robin Hasse

---

calcRenovationCostModel *Calculate linear renovation cost model*

Description

Specific renovation cost are estimated with a linear model with intercept w.r.t. to the renovation depth.

Usage

```
calcRenovationCostModel()
```

Details

The specific investment cost for renovation depending on the depth of renovation is calculated using a two-level statistical model fitted with data from a European Commission (EC) report on renovation. In a fist step, we fit a simple linear model with intercept to predict specific investment (USD/m2) with relative PE savings. In a second step, this linear model is scaled with one factor for each region. This scaling factor is predicted with GDP/POP using a negative exponential curve. For EU regions that are reported, we correct the result with a region-specific but time-invariant factor to best match the linear model to the data. This way, we get a linear model with intercept for each region that can be extrapolated to other regions (using GDP/POP) which still matches data of EU regions very well. The cost is finally disaggregated across residential building types (SFH, MFH) based on a rough cost factor seen in the ENTRANZE data.

Value

MAgPIE object with specific renovation cost

Author(s)

Robin Hasse

Source

https://op.europa.eu/s/xnYt

calcShareETP *Share of energy carriers or end uses in historic buildings demand*

Description

Shares are calculated based on energy demands in households and services from the IEA ETP data base. Missing shares that result from missing demand data are filled with the average share across all regions and periods and then normalised to sum up to one again. Biomass is split according to GDP per Capita (see toolSplitBiomass).

Usage

```
calcShareETP(subtype = c("enduse", "carrier"), feOnly = FALSE)
```

Arguments

`subtype`	specifies share
`feOnly`	specifies if shares or quantities are returned

Value

MAgPIE object with historic shares

Author(s)

Robin Hasse, Antoine Levesque, Hagen Tockhorn

calcShareOdyssee *Share of energy carriers or end uses in historic buildings demand*

Description

Shares are calculated based on energy demands in households and services from the Odyssee data base (EU member states). In case of multi-level shares, the function gives the share w.r.t. to the last categories but for all categories. E.g. 'enduse_carrier' gives the share of each carrier in the demand from each end use. Missing shares that result from missing demand data are filled with the average share across all regions and periods and then normalised to sum up to one again. Biomass is split according to GDP per Capita (see toolSplitBiomass).

Usage

```
calcShareOdyssee(
  subtype = c("enduse", "carrier", "enduse_carrier"),
  feOnly = FALSE
)
```

Arguments

`subtype`	Character, dimension names, level of shares.
`feOnly`	if TRUE, output is absolute FE values

Value

MAgPIE object with historic shares

Author(s)

Robin Hasse, Antoine Levesque, Hagen Tockhorn

`calcShares` *Historic carrier-Enduse Shares w.r.t Final Energy*

Description

Merges and transforms the calculated shares from the Datasets: Odyssee IEA_ETP TCEP WEO

Usage

```
calcShares(
  subtype = c("carrier_nonthermal", "carrier_thermal", "enduse_nonthermal",
    "enduse_thermal"),
  carrierCorrection = FALSE,
  feOnly = FALSE
)
```

Arguments

`subtype`	specifies share
`carrierCorrection`	allows additional corrections
`feOnly`	specifies if shares or quantities are returned

Details

Enduse shares are extrapolated using a linear regression where the ETP datapoint as well as selected regions from TCEP serves as origin and the entire TCEP dataset provides the growth factor.

Carrier shares are exported w.r.t. to all carriers per region per period. In the case of "carrierCorrection = TRUE", carrier shares are exported w.r.t. their contribution to each individual enduse. The reason for the latter is that the implemantation here is closer to EDGE-B, but didn't work well with further data processing steps (e.g. toolDisaggregate) where EC shares are not necessarily used. Hence, this case is by default disabled.

In the thermal case, the enduse "appliances" is transformed to "refrigerators" using the region-specific refrigerator share used in EDGE-B. Higher-resoluted data was not available.

Value

data.frame with historic energy demands

Note

The parameter "feOnly" is only applicable to IEA_ETP and TCEP data, since this is the necessary data to do a full disaggregation of EU and EC data, whereas Odyssee already gives disaggregated data for countries of the European Union.

As done in EDGE-B, enduse-disaggregated FE data from WEO for the regions MIE, AFR and JAP were used instead for values from IEA ETP.

Author(s)

Hagen Tockhorn, Robin Hasse

`calcSharesBuildingDemand` *Share of energy carriers or end uses in historic buildings demand*

Description

Shares are calculated based on energy demands from the Odyssee data base for EU member states,

Usage

```
calcSharesBuildingDemand(subtype = c("enduse", "carrier"))
```

Arguments

`subtype`	Character, dimension names, level of shares.

Value

MAgPIE object with historic shares

Author(s)

Robin Hasse, Antoine Levesque

`calcShareTCEP` *Calculate Enduse Shares from TCEP data base*

Description

Calculate Enduse Shares from TCEP data base

Usage

```
calcShareTCEP()
```

Value

magpie object

Author(s)

Hagen Tockhorn

`calcSurface` *calcSurface*

Description

Calculates country surface

Usage

```
calcSurface()
```

Value

Surface by country

Author(s)

Antoine Levesque

`calcUEdemand` *Calculate useful energy demand for space heating in buildings*

Description

Calculate useful energy demand for space heating in buildings

Usage

```
calcUEdemand()
```

Author(s)

Robin Hasse

`calcUValue` *Calculate U-value*

Description

Calculate U-value

Usage

```
calcUValue()
```

Author(s)

Robin Hasse

`cfac` *Calculate counterfactuals for solar radiation, wind speed, specific humidity and near-surface temperature as function of raster data on near-surface temperature.*

Description

The expected value of the respective climate variable (except temperature) is calculated from parameters taken from a preceding linear regression done in calcBAITpars where the respective variable is correlated with the near-surface atmospherical temperature. If no cell-resoluted parameters are given, the globally-meaned parameters from Staffel et. all 2023 are taken (see https://doi.org/10.1038/s41560-023-01341-5).

Usage

```
cfac(t, type, params = NULL)
```

Arguments

`t`	raster data on near-surface atmospherical temperature
`type`	considered climate variable
`params`	regression parameters as vector or raster object

Value

counterfactuals for respective climate variable

`checkDates` *Check if time period of BAIT input data (rsds, sfc, huss) is congruent with near-surface temperature data (tas).*

Description

Check if time period of BAIT input data (rsds, sfc, huss) is congruent with near-surface temperature data (tas).

Usage

```
checkDates(baitInput, tasData)
```

Arguments

`baitInput`	list of raster data encompassing different climate variables
`tasData`	raster data on near-surface atmosphere temperature

Value

baitInput with congruent time periods w.r.t. tasData

compBAIT *Calculate bias-adjusted internal temperature (BAIT)*

Description

BAIT is calculated from raster data on near-surface atmospherical temperature (tas), surface downdwelling shortwave radiation (rsds), near-surface wind speed (sfcwind) and near-surface specific humidity (huss). The latter three climate parameters are incorporated in the calculation of BAIT as the difference from their real value to the their expected value w.r.t. the near-surface temperature (see `cfac`). These are then incorporated in a weighted sum to account for the respective influence of each climate parameter on BAIT. The raster data containing BAIT is smoothed to account for the buildings' thermal inertia (see `smooth`) and blended with the near-surface temperature (see `blend`).

Usage

```
compBAIT(baitInput, tasData, weight = NULL, params = NULL)
```

Arguments

`baitInput`	named list containing rsds, sfcwind, huss climate data
`tasData`	raster data on near-surface atmospherical temperature
`weight`	named list with weights
`params`	optional named list with regression parameters from calcBAITpars

Value

raster object with BAIT values

compCellHDDCDD *Assign HDD/CDD values for given ambient/limit temperature*

Description

Assign HDD/CDD values for given ambient/limit temperature

Usage

```
compCellHDDCDD(temp, typeDD, tlim, factors)
```

Arguments

`temp`	raster data containing temperature/BAIT values
`typeDD`	type of degree day
`tlim`	limit temperature
`factors`	data frame with degree day values for `temp/tlim` combination

Value

raster object with HDD/CDD values

---

compHDDCDDFactors *Calculate HDD/CDD values for different ambient/limit temperature combinations*

Description

HDD/CDD values are pre-calculated for an interval `tlow-tup` and for a set of limit temperatures `tlim` with a temperature resolution of 0.1C.

Usage

```
compHDDCDDFactors(tlow, tup, tlim, tambStd = 5, tlimStd = 5)
```

Arguments

`tlow`	lower temperature boundary
`tup`	upper temperature boundary
`tlim`	named list of limit temperature sequences for `HDD` and `CDD`
`tambStd`	std of ambient temperature
`tlimStd`	std of limit temperature

Details

The respective heating/cooling degree days are calculated as the difference between the assumed ambient and a limit temperature, aggregated to a full day. The latter defines a threshold above/below which cooling/heating is assumed to be initiated.

To account for heterogenity in heating/cooling behavior, the ambient and limit temperature, `tamb` and `tlim`, are assumed to be normally distributed. This changes the calculation of a degree day to a double integration of `tLimit - T_ambient_day` with integration boundaries set at 3 standard deviations, `tambStd` and `tlimStd`, from `tamb` and `tlim` respectively.

As consequence, the ramp function of `HDD_day = max(0, tLimit - T_ambient_day)` changes to a curved function that is above zero even if the mean of `T_ambient_day` is above the mean of `tLimit`.

Value

data frame of HDD/CDD values

compStackHDDCDD *Calculate country-wise population-weighted HDD/CDD values*

Description

This function calculates country-wise population-weighted HDD/CDD values for raw ambient temperature or bias-adjusted internal temperature for a given set of limit temperatures from raster data on (various) climate variables.

Usage

```
compStackHDDCDD(
  ftas,
  tlim,
  countries,
  pop,
  factors,
  bait,
  frsds = NULL,
  fsfc = NULL,
  fhuss = NULL,
  wBAIT = NULL,
  params = NULL,
  rasDir = NULL
)
```

Arguments

`ftas`	file name of data on near-surface atmospherical temperature
`tlim`	named list of limit temperature sequences for `HDD` and `CDD`
`countries`	raster opbject defining (regional) aggregation boundaries
`pop`	raster object containing population data
`factors`	data frame with degree day values for `temp/tlim` combination
`bait`	boolean, BAIT is used as ambient temperature
`frsds`	file name of data on surface downdwelling shortwave radiation (optional)
`fsfc`	file name of data on near-surface wind speed (optional)
`fhuss`	file name of data on near-surface specific humidity (optional)
`wBAIT`	named list containing BAIT weights (optional)
`params`	raster object containing regression parameters from `calcBAITpars` (optional)
`rasDir`	absolute path to directory for saving raster files

Details

For further processing, raster objects containing degree day data are written for an interval of ten years.

`convertCensusHub` *convertCensusHub*

Description

Convert data from Census Hub

Usage

```
convertCensusHub(x, subtype)
```

Arguments

`x`	raw data
`subtype`	Eurostat code of data set

Value

MAgPie object with converted data

Author(s)

Robin Hasse

`convertDaioglou` *convertDaioglou*

Description

Tidy data from Daioglou et al. 2012. Currently, only the specific floor space (m2/cap) is available from household data file. UN sources are dropped, if there is data from other sources for the same region. For the remaining data, if there is data from multiple UN Source for a region and period (i.e. different cities), the average of those sources is considered. This neglects the differences in population size of the cities. Selected data points are removed additionally.

Usage

```
convertDaioglou(x, subtype = "households.specific floor space")
```

Arguments

`x`	MAgPIE object with data from Daioglou et al. #nolint
`subtype`	.

Value

clean MAgPIE object with unique data points

Author(s)

Robin Hasse, Antoine Levesque

`convertDeetman2020` *Convert Weibull lifetime distribution parameters for buildings*

Description

Convert Weibull lifetime distribution parameters for buildings

Usage

```
convertDeetman2020(x, subtype)
```

Arguments

`x`	MAgPIE object with data from Deetman et al. 2020
`subtype`	character, building subsector (either 'residential' or 'commercial')

Author(s)

Robin Hasse

`convertEEAfloorspace` *convert EEAfloorpace*

Description

Convert historic floor space data from the European Environment Agency.

Usage

```
convertEEAfloorspace(x)
```

Arguments

`x`	MAgPIE object with floor space data

Value

floor national floor space in million m2

Author(s)

Robin Hasse

convertEnergiforsk2016 *Convert European district heating prices*

Description

We should consider prper extrapolation

Usage

```
convertEnergiforsk2016(x)
```

Arguments

`x`	MAgPIE object with original district heating prices

Author(s)

Robin Hasse

convertEUBuildingsDB *convertEUBuildingsDB*

Description

Convert to common units.

Usage

```
convertEUBuildingsDB(x, subtype)
```

Arguments

`x`	MAgPIE object with data from EU Buildings Database
`subtype`	character .

Value

clean MAgPIE object with

Author(s)

Robin Hasse

`convertEurObservER` *convertEurObservER*

Description

Each report of the Heat Pumps Barometer covers the two years before. There numerous cases, where reports contradict in overlapping values or there are huge jumps (probably due to changes in the accounting). This function creates smooth trajectories back-casting from the most recent data. If data overlaps, earlier data is scaled to match more recent data. In case of gaps, the trajectories are back-casted maintaining relative growth within each report. Growth rates between reports are interpolated. The Method yields mostly smooth trajectories always hitting the mos recent reported numbers.

Usage

```
convertEurObservER(x, subtype)
```

Arguments

`x`	raw data
`subtype`	Eurostat code of data set

Value

MAgPie object with converted data

Author(s)

Robin Hasse

`convertEuropeanCommissionRenovation` *convertEuropeanCommissionRenovation*

Description

Drop EU28 aggregate and rename regions to ISO3.

Usage

```
convertEuropeanCommissionRenovation(x)
```

Arguments

`x`	MAgPIE object with data from EC report

Value

clean MAgPIE object with data from EU Buildings Database

Author(s)

Robin Hasse

---

convertEurostatBuildings *convertEurostatBuildings*

Description

Convert buildings-related data from Eurostat.

Usage

```
convertEurostatBuildings(x, subtype)
```

Arguments

`x`	raw data
`subtype`	Eurostat code of data set

Value

MAgPie object with converted data

Author(s)

Robin Hasse

---

convertGDL *convertGDL*

Description

Drop subnational data but keep other subgroups (i.e. rural/urban, income quartiles)

Usage

```
convertGDL(x, subtype)
```

Arguments

`x`	MAgPIE object with data from EU Buildings Database
`subtype`	.

Value

clean MAgPIE object with data from EU Buildings Database

Author(s)

Robin Hasse

`convertHDDCDD` *convert HDDCDD*

Description

Convert historic and scenario based HDD CDD data to data on ISO country level.

Usage

```
convertHDDCDD(x)
```

Arguments

`x`	MAgPIE object containing HDD CDD values at ISO country resolution

Value

HDD CDD data as MAgPIE object aggregated to country level

Author(s)

Antoine Levesque

convertHotmaps *convert Hotmaps*

Description

convert Hotmaps

Usage

```
convertHotmaps(x)
```

Arguments

`x`	MagPIE object with raw building stock data

Value

MagPIE object with converted data

Author(s)

Robin Hasse

convertISIMIPbuildings *convert ISIMIP data for mredgebuildings*

Description

convert ISIMIP data for mredgebuildings

Usage

```
convertISIMIPbuildings(x, subtype)
```

Arguments

`x`	MAgPIE object on cellular level
`subtype`	filename

Value

rasterBrick object

Author(s)

Hagen Tockhorn

`convertOdyssee` *convertOdyssee*

Description

Rename regions and convert unit

Usage

```
convertOdyssee(x, subtype = "households")
```

Arguments

`x`	MAgPIE object with data from Odyssee Database
`subtype`	category

Value

clean MAgPIE object

Author(s)

Robin Hasse

`convertPFUDB` *Converts Data from Primary Fuel and Useful Energy Data Base (PFUDB)*

Description

This was adapted from EDGE function 'getPFUDB.R'.

Usage

```
convertPFUDB(x)
```

Arguments

`x`	MAgPIE object with data from PFUDB #nolint

Value

magpie object

Author(s)

Hagen Tockhorn

convertTCEP *Convert TCEP data base*

Description

Convert TCEP data base

Usage

```
convertTCEP(x)
```

Arguments

x	MAgPIE object with data from TCEP #nolint

Value

magpie object

Author(s)

Hagen Tockhorn, Robin Hasse

convertUNHouseholds *convertUNHouseholds*

Description

Clean up UN data on household size and composition

Usage

```
convertUNHouseholds(x, subtype)
```

Arguments

x	MAgPIE object with data from EU Buildings Database
subtype	household variable

Value

clean MAgPIE object with

Author(s)

Robin Hasse

`convertWEO` *Convert WEO enduse shares w.r.t. to global final energy demand*

Description

Convert WEO enduse shares w.r.t. to global final energy demand

Usage

```
convertWEO(x)
```

Arguments

`x`	readWEO object

Author(s)

Hagen Tockhorn

`extrapolateMissingPeriods` *Extrapolate missing values beyond existing periods*

Description

Extrapolate missing values beyond existing periods

Usage

```
extrapolateMissingPeriods(chunk, key, slopeOfLast = 5)
```

Arguments

`chunk`	grouped data.frame
`key`	column holding values for extrapolation
`slopeOfLast`	number of values for boundary regression

Value

data.frame with extrapolated column `key`

Author(s)

Robin Hasse, Hagen Tockhorn

fullBRICK *fullBRICK*

Description

Compute complete input data set for BRICK

Usage

```
fullBRICK(rev = 0)
```

Arguments

rev	data revision which should be used as input (positive numeric).

Author(s)

Robin Hasse

fullEDGEBUILDINGS *fullEDGEBUILDINGS*

Description

Function that produces the complete ISO data set required for the EDGE - Buildings model.

Usage

```
fullEDGEBUILDINGS(rev = 0)
```

Arguments

rev	data revision which should be used as input (positive numeric).

Author(s)

Antoine Levesque, Robin Hasse, Hagen Tockhorn

See Also

readSource,getCalculations,calcOutput

Examples

```
## Not run:
fullEDGEBUILDINGS()

## End(Not run)
```

getFEbyEUEC *Historic Final Energy Demand by Carrier and Enduse*

Description

Takes the historic final energy demand by carriers from IEA and disaggregates it into different end uses.

Usage

```
getFEbyEUEC()
```

Value

data.frame with historic energy demands

Author(s)

Hagen Tockhorn

join_all *Join by all common columns*

Description

Additional columns to join by that have different names can be give. The kind of joint can also be specified.

Usage

```
join_all(
  x,
  y,
  by = NULL,
  .direction = "left",
  exclude = "value",
  silent = TRUE,
  ...
)
```

Arguments

`x, y`	A pair of data frames
`by`	character vector of columns to join by beyond the commonly named
`.direction`	type of dplyr join, 'left' by default
`exclude`	character vector with column names that should be excluded from the search for join columns
`silent`	boolean, whether to print type of joint the join columns
`...`	additional arguments passed to join function

Author(s)

Robin Hasse

`prepBaitInput` *Read in necessary climate data to calculate BAIT or calculate mean values of said climate data to fill missing data in case of temporal mismatch between near-surface atmospherical temperature and other considered climate data.*

Description

Read in necessary climate data to calculate BAIT or calculate mean values of said climate data to fill missing data in case of temporal mismatch between near-surface atmospherical temperature and other considered climate data.

Usage

```
prepBaitInput(
  frsds = NULL,
  fsfc = NULL,
  fhuss = NULL,
  baitInput = NULL,
  fillWithMean = FALSE
)
```

Arguments

`frsds`	file path to raster data on surface downdwelling shortwave radiation
`fsfc`	file path to raster data on near-surface wind speed
`fhuss`	file path to raster data on near-surface specific humidity
`baitInput`	named list of climate data
`fillWithMean`	boolean, only mean is calculated and returned

Value

named list with read-in or meaned climate data

`readCensusHub` *Read data from the European Census Hub*

Description

The Census Hub of the European Statistical System provides census data from 2011 on population and housing of EU member states. 2021 census data is announced to be published there too.

- typeVintage: number of dwellings of different construction periods and building types (single, two, multiple dwellings). Show data on: dwellings, location: nations, Topics: Type of building, Period of construction (select all) -> CSV (Separator: Comma)

Usage

```
readCensusHub(subtype)
```

Arguments

`subtype`	census subset

Value

MAgPIE object with data

Author(s)

Robin Hasse

Source

https://ec.europa.eu/CensusHub2/

`readDaioglou` *Household and demand data from Daioglou et al. 2012*

Description

Compilation of survey data. Quintile 0 stands for the entire population.

- households: Global data on population, household size, floor space per capita (m2/cap), population density (cap/km2), GINI and household expenditures (PPP2005/cap). Sources are written into a dimension to allow later cleaning based on sources
- shares: Final energy shares of end uses in buildings. Sources are dropped.

Usage

```
readDaioglou(subtype = "households.specific floor space")
```

Arguments

`subtype` .

Author(s)

Robin Hasse, Antoine Levesque

References

https://doi.org/10.1016/j.energy.2011.10.044

`readDeetman2020` *Read Weibull lifetime distribution parameters for buildings*

Description

Read Weibull lifetime distribution parameters for buildings

Usage

```
readDeetman2020(subtype = "residential")
```

Arguments

`subtype` character, building subsector (either 'residential' or 'commercial')

Value

MagPIE object with Weibull lifetime parameters for different world regions

Author(s)

Robin Hasse

Source

https://doi.org/10.1016/j.jclepro.2019.118658

readECEMF *Read ECEMF data*

Description

Project data: European Climate and Energy Modelling Forum

Usage

```
readECEMF(subtype = "FEPrices")
```

Arguments

subtype	character, type of data

Details

- FEPrices: price components w/o VAT in EUR/MWh

Author(s)

Robin Hasse

readEEAfloorspace *Read national floor space from EEA*

Description

Floor space for many European countries between 1990 and 2009 in thousand m2.

Usage

```
readEEAfloorspace()
```

Value

magpie object with floor space

Author(s)

Robin Hasse

See Also

readSource

Examples

```
## Not run:  a <- readSource(type = "EEAfloorspace")
```

---

readEHI *Read EHI Heating Market Report*

Description

Read EHI Heating Market Report

Usage

```
readEHI(subtype)
```

Arguments

`subtype`	Character, report to read

Value

magpie object

Author(s)

Robin Hasse

---

readEnergiforsk2016 *Read European district heating prices*

Description

Read European district heating prices

Usage

```
readEnergiforsk2016()
```

Author(s)

Robin Hasse

`readEUBuildingsDB` *Read EU Buildings Database*

Description

Compilation of various data sources from the European Commission. Because of the considereable size of this database, a subtype has to be provided. The subtype represents the category of items. It is possible to append selected variables for further filtering.

Usage

```
readEUBuildingsDB(subtype = "")
```

Arguments

`subtype`	character .

Value

magpie object

Author(s)

Robin Hasse

See Also

`readSource`

Examples

```
## Not run:  a <- readSource(type = "EUBuildingsDB")
```

`readEurObservER` *Read EurObservER*

Description

PDF market reports on Renewables in Europe

Usage

```
readEurObservER(subtype)
```

Arguments

`subtype`	.

Value

magpie object

Note

I download the report and snip a png of the table. Then I extract the data with this online tool and save it as a tabular csv in the source folder. The Tool is not fully reliable. This function throws an error if values don't sum up. You then have to correct the csv file manually.

Author(s)

Robin Hasse

Source

https://www.eurobserv-er.org/

`readEuropeanCommissionRenovation`

EU report on historic renovation

Description

This function reads data on renovation from a pdf file of an EC report from 2019 (Tables 2 - 13). There is no period dimension. The data represents averages between 2012 and 2016.

Usage

```
readEuropeanCommissionRenovation(subtype = "csv")
```

Arguments

`subtype`	character, source to read from: `"csv"` (default) or `"pdf"` (requires the suggested package pdftools)

Value

MAgPIE object with renovation data

Author(s)

Robin Hasse

Source

https://op.europa.eu/en/publication-detail/-/publication/97d6a4ca-5847-11ea-8b81-01aa75ed71a1

See Also

`readSource`

Examples

```
## Not run:  a <- readSource(type = "EuropeanCommissionRenovation")
```

`readEurostatBuildings` *Read Eurostat data on buildings*

Description

Read various data sets from Eurostat that are used in the modelling of buildings

- nrg_inf_hptc: Heat pumps - technical characteristics by technologies
- ilc_hcmh02: Average size of dwelling by household type and degree of urbanisation
- nrg_d_hhq: Disaggregated final energy consumption in households

Usage

```
readEurostatBuildings(subtype)
```

Arguments

`subtype`	Eurostat code of data set

Value

MAgPIE object with data

Note

see https://ec.europa.eu/eurostat/web/energy/data/energy-balances for definitions of codes

Author(s)

Robin Hasse

readGDL *Read Global Data Lab*

Description

Different databases each comprising the harmonised results of regional/national surveys. Available subtypes are

- SHDI: Subnational Human development Database
- AreaDatabase: Subnational development Indicators (low/medium income countries)

Usage

```
readGDL(subtype)
```

Arguments

`subtype` .

Value

magpie object

Author(s)

Robin Hasse

Source

https://globaldatalab.org/

References

https://doi.org/10.1038/sdata.2019.38

https://globaldatalab.org/asset/286/Smits%20GDL%20Working% 20Paper%2016-101%20v360.pdf

See Also

readSource

Examples

```
## Not run:  a <- readSource(type = "GDL")
```

readHDDCDD *Read HDD CDD*

Description

Read heating and cooling degree days; past and scenario data as magclass objects

Usage

```
readHDDCDD(subtype = 18)
```

Arguments

subtype	Temperature threshold for computing HDD and CDD

Value

magpie object HDD CDD

Author(s)

Antoine Levesque

See Also

readSource

Examples

```
## Not run:  a <- readSource(type="HDDCDD")
```

readHotmaps *read EU28 building stock from Hotmaps*

Description

Consistent building stock of EU28 residential and commercial sector. The data presents a snapshot, I guess in 2016 but this is not clear to me yet.

Usage

```
readHotmaps()
```

Details

There are a few duplicate rows. Due to inconsistent values we average across duplicates.

Author(s)

Robin Hasse

`readIEAfloorspace` *Floor space by subsectors*

Description

Residential and commercial floor space for big world regions from the IEA TCEP report 2014 in billion m2.

Usage

```
readIEAfloorspace()
```

Value

MAgPIE object with floor space

Author(s)

Robin Hasse

Source

https://iea.blob.core.windows.net/assets/416e9555-67f9-49ed-95ed-8afccd71b433/ Tracking_clean_energy_progress_2014.p

See Also

readSource

Examples

```
## Not run:  a <- readSource(type = "IEAfloorspace")
```

`readISIMIPbuildings` *Read relevant ISIMIP data for mredgebuildings*

Description

Relevant data such as region masks, population and relevant climate data are read in. The relevant file is declared in the subtype with the full file name.

Usage

```
readISIMIPbuildings(subtype)
```

Arguments

`subtype`	filename

Details

If the file name includes a suffix in the form of an integer such as `_<int>.filetype`, the file is split into a single year period, e.g. `<filename>_2001_2010_2.nc` will return data for the second year of the 2001-2010 period, here 2002.

Note

folder structure in inputdata/sources/ISIMIPbuildings is expected to be: country masks : var/ population : var/scenario other : var/scenario/model

currently, this function only reads data from ISIMIP3b

Author(s)

Hagen Tockhorn

`readOdyssee` *Read Odyssee data base*

Description

Copied from website: The Odyssee database contains detailed energy consumption by end-use and their drivers as well as energy efficiency and CO2-related indicators. Latest available data is provided by national representatives, such as energy agencies or statistical organization, from all EU countries as well as Norway, Serbia, Switzerland and the United Kingdom.

Usage

```
readOdyssee(subtype = "households")
```

Arguments

`subtype`	database category

Value

magpie object

Note

To download new data, log into the website, select all items of all levels and download the data 'column-orientated csv'.

Variables are labels with the item code but full names can be found in the source data

Author(s)

Pascal Führlich, Robin Hasse

Source

https://odyssee.enerdata.net/database/

`readPFUDB` *Read Data from Primary Fuel and Useful Energy Data Base (PFUDB)*

Description

This was adapted from EDGE function 'getPFUDB.R'.

Usage

```
readPFUDB()
```

Value

magpie object

Author(s)

Hagen Tockhorn

`readTCEP` *Read TCEP data base*

Description

Read TCEP data base

Usage

```
readTCEP()
```

Value

magpie object

Author(s)

Hagen Tockhorn

`readUNHouseholds` *Read UN data on Household Size & Composition*

Description

Compilation of indicators on household size and membership composition. The estimates are based on 814 unique data points from 172 countries or areas, representing approximately 97 per cent of the world's population in 2019, collected between 1960 to 2018.

Usage

```
readUNHouseholds(subtype)
```

Arguments

`subtype`	household variable

Value

magpie object

Author(s)

Robin Hasse

Source

https://www.un.org/development/desa/pd/data/household-size-and-composition

See Also

readSource

Examples

```
## Not run:  a <- readSource(type = "UNHouseholds")
```

---

readWEO *Read WEO enduse shares w.r.t. to global final energy demand*

Description

Read WEO enduse shares w.r.t. to global final energy demand

Usage

```
readWEO(subtype)
```

Arguments

subtype	variable to define data subset ("Buildings", "Transport", "Industry")

Author(s)

Hagen Tockhorn

---

smooth *Smooth data over preceding two days*

Description

Smooth data over preceding two days

Usage

```
smooth(r, weight)
```

Arguments

r	raster data to be smoothed
weight	named list with smoothing parameter sig

Value

smoothed raster data

toolAddThermal *Introduce refrigerators as appliances*

Description

Introduce refrigerators as appliances

Usage

```
toolAddThermal(df, mapping, fridgeShare, feOnly = TRUE, shareOf = NULL)
```

Arguments

`df`	data.frame
`mapping`	original EDGE-B region mapping
`fridgeShare`	regional shares of refrigerators
`feOnly`	are absolute values (TRUE) or shares (FALSE) considered?
`shareOf`	column name to calculate share of - must be given if feOnly = FALSE

Author(s)

Hagen Tockhorn

toolCalcShares *Calculate Shares*

Description

Calculate Shares

Usage

```
toolCalcShares(df, colShare)
```

Arguments

`df`	data.frame
`colShare`	column to be grouped over

Value

data.frame with calculated shares for colShare

Author(s)

Hagen Tockhorn

toolCountryFillAvg *Tool: CountryFillAvg*

Description

This function expects a MAgPIE object with ISO country codes in the spatial dimension. It applies toolCountryFill but fills with the average across regions.

Usage

```
toolCountryFillAvg(x, ...)
```

Arguments

x	MAgPIE object with ISO country codes in the spatial dimension
...	arguments passed to toolCountryFill

Author(s)

Robin Hasse

toolDisaggregate *Disaggregate energy demand by carriers into end uses by each carrier*

Description

This function takes energy demand data that differentiates only by carrier and disaggregates it further such that all relevant combinations of carrier and end use are differentiated. It ensures that given end use shares (across all carriers) are met.

Usage

```
toolDisaggregate(
  data,
  enduseShares,
  outliers = NULL,
  exclude = NULL,
  dataDisagg = NULL,
  regionMapping = NULL
)
```

Arguments

`data`	data.frame with energy demand data differentiated by carriers that is to be disaggregated
`enduseShares`	data.frame with end use shares that have to be met. These shares might be given at aggregated regional resolution if `regionmapping` is provided.
`outliers`	list of regions where naive disaggregation estimate shall be used.
`exclude`	data.frame with the columns `carrier` and `enduse` that should list all combinations of the two that are excluded. All remaining combinations of the carriers in `data` and the end uses in `enduseShares` are considered.
`dataDisagg`	data.frame similar to `data` but already disaggregated by carriers and end uses. The average distribution of its disaggregation will be used as the target distribution for the minisation.
`regionMapping`	data.frame with the columns `region` and `regionAgg` that maps the regions between `data` and `enduseShares`.

Details

The function essentially performs a quadratic optimisation. The constraints make sure that both the carrier quantities and the end use shares are met. This generally leaves infinite solutions (in problematic cases none). To get to one disaggregation, the deviation from a given distribution across all relevant combinations of carriers and end uses is minimised.

Author(s)

Hagen Tockhorn, Robin Hasse

`toolSplitBiomass` *Splits Biomass according to GDP per Capita*

Description

This is adapted from EDGE-B Function "splitBiomass.R" by Antoine Levesque.

Usage

```
toolSplitBiomass(df, dfGDPpop, min = 10000, max = 15000, varName = "biomass")
```

Arguments

`df`	dataframe with carrier as 'variable'
`dfGDPpop`	dataframe containing the GDP per Capita
`min`	lower threshold of GDP per Capita in USD 2005
`max`	upper threshold of GDP per Capita in USD 2005
`varName`	nomenclature of biomass

Details

Any value belonging to the carrier defined by "varName" will be split into the default carrier names "biomod" and "biotrad" according to weights given by "dfGDPpop".

Value

factor.data.frame

Author(s)

Hagen Tockhorn

`toolUnitConversion` *Convert unit*

Description

Takes MAgPIE object with a variable dimension that has the unit as suffix separated with '_' and converts the unit according to a unit conversion table.

Usage

```
toolUnitConversion(x, unitConversion, dim = 3.1, removeUnit = FALSE)
```

Arguments

`x`	MAgPIE object
`unitConversion`	data.frame with the columns from, to, factor
`dim`	variable dimension that contains the unit as suffix
`removeUnit`	boolean remove unit after conversion for clean variable names

Author(s)

Robin Hasse

`usd2eur` *Exchange rate between USD and EUR*

Description

Exchange rate between USD and EUR

Usage

```
usd2eur(year = 2020)
```

Arguments

`year`	integer, reference year

Value

MER EUR in USD in given year

Author(s)

Robin Hasse

Index