Package 'ejscreen'

June 17, 2016

Type Package

Title EJSCREEN Tools for US EPA Environmental Justice Mapping and Screening

Version 0.1

Date 2015-07-19

Author info@ejanalysis.com

Maintainer ejanalyst <info@ejanalysis.com>

Description Data and tools related to the United States Environmental Protection Agency's screening and mapping tool for environmental justice, EJSCREEN

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 3.1.2),
Hmisc,
proxistat,
ejanalysis,
analyze.stuff

URL http://ejanalysis.github.io
http://www.ejanalysis.com/
http://www.epa.gov/ejscreen

RoxygenNote 5.0.1

R topics documented:

`download.ejscreen` *Download the EJSCREEN Dataset for use in R*

Description

Download EJSCREEN dataset from FTP site, and import to R as data.table, optionally adding a flag field.

Usage

```
download.ejscreen(folder = getwd(),
  ftpurl = "ftp://newftp.epa.gov/EJSCREEN",
  zipname = "EJSCREEN_20150505.csv.zip", csvname = "EJSCREEN_20150505.csv",
  addflag = FALSE, cutoff = 80, or.tied = TRUE)
```

Arguments

`folder`	Optional path to folder (directory) where the file will be downloaded and unzipped. Default is current working directory.
`ftpurl`	Optional. Default is 'ftp://newftp.epa.gov/EJSCREEN' for where to find the zipped data.
`zipname`	Optional. Default is 'EJSCREEN_20150505.csv.zip' for the name of the zip file of data.
`csvname`	Optional. Default is 'EJSCREEN_20150505.csv' for the name of the csv file that was zipped.
`addflag`	Optional. Default is FALSE. If TRUE, it adds a field called flagged, which is TRUE if 1 or more of the EJ Indexes is at/above the cutoff US percentile.
`cutoff`	Optional. Default is 80. See addflag parameter.
`or.tied`	Optional. Default is TRUE, meaning at or above the cutoff. FALSE means above only. See addflag parameter.

Details

Designed for 2015 version only right now. Not tested.

Value

Returns a data.frame with ejscreen dataset of environmental and demographics indicators, and EJ Indexes, as raw values, US percentiles, text for popups. Output has one row per block group.

Source

See http://www.epa.gov/ejscreen for more information, and see http://www.epa.gov/ejscreen/download-ejscreen-data or ftp://newftp.epa.gov/EJSCREEN for raw data.

See Also

ejscreen.create

Examples

```
## Not run:

 bg <- download.ejscreen(folder='~', addflag=TRUE)

## End(Not run)
```

---

ejscreen *Tools for EJSCREEN, US EPA's Environmental Justice (EJ) Screening and Mapping Tool*

Description

This R package provides tools related to environmental justice (EJ) analysis, specifically related to the United States Environmental Protection Agency (EPA) screening and mapping/GIS tool called EJSCREEN. See http://www.epa.gov/ejscreen This package facilitates development of the EJSCREEN dataset, based on user-provided environmental indicators. The resulting dataset is a data.frame that contains data on demographics (e.g., percent of residents who are low-income) and user-provided local environmental indicators (e.g., an air quality index), and calculated indicators called EJ Indexes, which combine environmental and demographic indicators. The dataset also provides each key indicator as a national population-percentile that represents what percentage of the US population have equal or lower raw values for the given indicator. The dataset has one row per spatial location (e.g., Census block group).

Details

Key functions include

- ejscreen.create
- ejscreen.lookuptables
- Various functions from the **ejanalysis** package are also relevant.

References

http://ejanalysis.github.io
http://www.ejanalysis.com/
http://www.epa.gov/ejscreen

`ejscreen.acs.calc` *Create Calculated EJSCREEN Variables*

Description

Use specified formulas to create calculated, derived variables such as percent low income. Relies upon `calc.fields` from **analyze.stuff** package.

Usage

```
ejscreen.acs.calc(bg, folder = getwd(), keep.old, keep.new, formulafile,
  formulas)
```

Arguments

`bg`	Data.frame of raw demographic data counts, and environmental indicators, for each block group, such as population or number of Hispanics.
`folder`	Default is getwd(). Specifies path for where to read from (if formulafile specified) and write to.
`keep.old`	Vector of variables names from names(bg), indicating which to return (retain, not drop). Default is to keep only the ones that match the list of default names in this code. Or this can be simply 'all' which means keep all input fields.
`keep.new`	Vector of variables names of new created variables, indicating which to return (retain, not drop). Default is to keep a specific list of fields (see source code). Or this can be simply 'all' which means keep all new fields.
`formulafile`	Name of optional csv file with column called formula, providing R syntax formulas as character fields. If not specified, function loads this as data(ejscreenformulas). Example of one formula: 'pctunder5 <- ifelse(pop==0,0, under5/pop)' Use a result of zero in cases where the denominator is zero, to avoid division by zero. For example, the formula `'pctmin <- ifelse(pop==0,0, as.numeric(mins ) / pop)'` indicates that percent minority is calculated as the ratio of number of minorities over total population of a block group, but is set to zero if the population is zero.
`formulas`	Options vector of formulas as character strings that contain R statements in the form "var1 <- var2 + var3" for example. Either formulafile or formulas can be specified (or neither) but not both (error). Formulas should be in the same format as a formulafile field or the contents of ejscreenformulas (via data(ejscreenformulas) or lazy loading like x <- ejscreenformulas).

Value

Returns a data.frame with some or all of input fields (those in keep.old), plus calculated new fields (those in keep.new).

Examples

```
set.seed(99)
 envirodata=data.frame(FIPS=analyze.stuff::lead.zeroes(1:1000, 12),
   air=rlnorm(1000), water=rlnorm(1000)*5, stringsAsFactors=FALSE)
 demogdata=data.frame(FIPS=analyze.stuff::lead.zeroes(1:1000, 12),
   pop=rnorm(n=1000, mean=1400, sd=200), mins=runif(1000, 0, 800),
```

```
  num2pov=runif(1000, 0,500), stringsAsFactors=FALSE)
demogdata$povknownratio <- demogdata$pop
x=ejscreen.acs.calc(bg=demogdata)
```

`ejscreen.acs.rename` *Rename Fields of ACS Data for Use in EJSCREEN*

Description

Start with raw counts from demographic survey data, and environmental data, and rename fields to use friendly variable names.

Usage

```
ejscreen.acs.rename(acsraw, folder = getwd(), formulafile)
```

Arguments

`acsraw`	Data.frame of raw data counts for each block group, such as population or number of Hispanics.
`folder`	Default is getwd(). Specifies path for where to read from (if formulafile specified) and write to.
`formulafile`	Default if this is blank is to use data(ejscreenformulas). Otherwise filename must be specified. If not specified, function loads this as data().

Value

Returns a data.frame with some or all of input fields, plus calculated new fields.

Examples

```
# (no examples yet)
```

`ejscreen.create` *Create EJSCREEN Dataset from Environmental Indicators*

Description

Start with raw environmental indicator data, and create full EJSCREEN dataset. This code also contains an outline of steps involved.

Usage

```
ejscreen.create(e, acsraw, folder = getwd(), keep.old, formulas,
  mystates = "all", demogvarname0 = "VSI.eo", demogvarname1 = "VSI.svi6",
  wtsvarname = "pop", checkfips = TRUE, EJprefix0 = "EJ.DISPARITY",
  EJprefix1 = "EJ.BURDEN", EJprefix2 = "EJ.PCT",
  ejformulasfromcode = FALSE, ejtype = 1, demogvarname0suffix = "eo",
  demogvarname1suffix = "svi6", end.year, threshold = FALSE, cutoff = 0.8,
  thresholdfieldnames)
```

Arguments

e	Data.frame of raw data for environmental indicators, one row per block group, one column per indicator.
acsraw	Optional data.frame of raw demographic indicators. Downloaded if not provided as parameter.
folder	Optional, default is getwd(). Passed to get.acs if demog data must be downloaded. Passed to but not currently used by ejscreen.acs.rename which uses change.fieldnames in **analyze.stuff** package. Not currently passed to ejscreen.acs.calc which uses calc.fields in **analyze.stuff** package.
keep.old	optional vector of colnames from e that are to be used/returned. For nondefault colnames, this must be used.
formulas	optional, see ejscreen.acs.calc for details. Defaults are in ejscreenformulas$formula Note that if formulas is specified, ejformulasfromcode is ignored.
mystates	optional vector of 2-letter state abbreviations. Default is "all" which specifies all states plus DC plus PR.
demogvarname0	optional, default is 'VSI.eo' used as demographic indicator for EJ Indexes. Must be a colname in acsraw or created and kept by formulas.
demogvarname1	optional, default is 'VSI.svi6' used for alternative EJ Indexes. Must be a colname in acsraw or created and kept by formulas.
wtsvarname	optional, default is 'pop' used for weighted percentiles, etc. Must be a colname in acsraw or created and kept by formulas.
checkfips	optional, default is TRUE. If TRUE, verifies all FIPS are valid. To use something other than actual US FIPS codes, set this to FALSE.
EJprefix0	optional, default is 'EJ.DISPARITY' - specifies prefix for colnames of main EJ Indexes, with a period separating prefix from body of colname
EJprefix1	optional, default is 'EJ.BURDEN' - specifies prefix for colnames of Alternative 1 version of EJ Indexes, with a period separating prefix from body of colname
EJprefix2	optional, default is 'EJ.PCT' - specifies prefix for colnames of Alternative 2 version of EJ Indexes, with a period separating prefix from body of colname
ejformulasfromcode	optional, default is FALSE. If TRUE, use EJ Index formulas built into this function instead of the EJ Index formulas in ejscreenformulas. The parameters such as demogvarname0 are only used if ejformulasfromcode=TRUE. Note that if formulas is specified, ejformulasfromcode is ignored.
ejtype	optional, default is 1, defines which formula to use for ejindex if not using ejscreenformulas. See ej.indexes But note alt1 and alt2 still use type 5 and 6 ignoring ejtype.
demogvarname0suffix	optional, default is 'eo' - specifies suffix for colnames of EJ Indexes based on demogvarname0, with a period separating body of colname from suffix
demogvarname1suffix	optional, default is 'svi6' - specifies suffix for colnames of EJ Indexes based on demogvarname1, with a period separating body of colname from suffix
end.year	optional to pass to get.acs (such as end.year='2013' – otherwise uses default year used by get.acs)
threshold	optional, default is FALSE. Set to TRUE to add a column (called 'flag') to results that is TRUE when one or more of certain percentiles (US EJ Index) in a block group (row) exceed cutoff.

cutoff — optional, default is 0.80 (80th percentile). If threshold=TRUE, then cutoff defines the threshold against which percentiles are compared.

thresholdfieldnames — optional, default is standard EJSCREEN EJ Indexes built into code. Otherwise, vector of character class fieldnames, specifying which fields to compare to cutoff if threshold=TRUE.

checkfips — optional, default is TRUE. If TRUE, function checks to verify all FIPS codes appear to be valid US FIPS (correct number of characters, adding any leading zero needed, and checking the first five to ensure valid county)

Details

**Note that if non-default fieldnames are used in e and/or acsraw, those must be specified in parameters including demogvarname0, demogvarname1, wtsvarname, keep.old (and could be reflected in prefix and suffix params as well).

Value

Returns a data.frame with full ejscreen dataset of environmental and demographics indicators, and EJ Indexes, as raw values, US percentiles, and text for popups. Output has one row per block group.

Examples

```
## Not run:
 set.seed(99)
 envirodata=data.frame(FIPS=analyze.stuff::lead.zeroes(1:1000, 12),
   air=rlnorm(1000), water=rlnorm(1000)*5, stringsAsFactors=FALSE)
 demogdata=data.frame(FIPS=analyze.stuff::lead.zeroes(1:1000, 12),
   pop=rnorm(n=1000, mean=1400, sd=200), mins=runif(1000, 0, 800),
   num2pov=runif(1000, 0,500), stringsAsFactors=FALSE)
 demogdata$povknownratio <- demogdata$pop
 # downloads ACS demographics and combines with user provided envirodata:
 # bg1=ejscreen.create(envirodata, mystates=c('de','dc'))
 # currently does not work for nonstandard colnames
 #  unless keep.old used as follows (work in progress):
 y=ejscreen.create(e=envirodata, acsraw=demogdata,
   keep.old = c(names(envirodata), names(demogdata)),
   demogvarname0 = 'pctmin', demogvarname1 = 'pctlowinc', wtsvarname = 'pop' )

## End(Not run)
```

---

ejscreen.lookuptables *Create EJSCREEN Lookup Tables of Pop. Percentiles by Zone*

Description

*** Work in progress as of mid 2015. Start with raw environmental, demographic, and EJ indicator data, and write as csv files to disk a series of lookup tables that show population percentiles and mean values for each indicator.

Usage

```
ejscreen.lookuptables(x, weights = x$pop, cols, zonecols = c("ST",
  "REGION"), folder = getwd(), missingcode = NA)
```

Arguments

x	Data.frame of indicators, one row per block group, one column per indicator.
weights	Weights for percentiles – Default is population count to provide population percentiles.
cols	Optional vector of colnames of x that need percentile lookup tables, or 'all' which means all numeric fields in x. Default is a standard set of EJSCREEN fieldnames defined within this function (see source code).
zonecols	Optional. Must set to NULL if no zones wanted, because default is c('ST', 'REGION'), names of cols in x that contain zone codes, such as State names or Region numbers, used to create a lookup table file for each of the zonecols, with separate percentiles calculated within each zone.
folder	Default is getwd() - specifies where to save the csv files.
missingcode	Leave this unspecified if missing values are set to NA in the input data. Default is -9999999 (but if already NA then do not specify anything for this). The number or value in the input data that designates a missing value.

Details

Percentiles are calculated as exact values and then rounded down to the nearest 0-100 percentile. This calculates percentiles among only the non-NA values. In other words, people in places with missing data are excluded from the calculation. This means the percentile is the percent of people with valid data (i.e., not NA) who have a tied or lower value.

Value

Overall lookup table(s) as data.frame (but not zonal ones). Creates lookup tables saved as csv files to specified folder. One table for overall percentiles, and one for each of the zonecols (unless that is set to NULL).

Examples

```
## Not run:
 set.seed(99)
 envirodata=data.frame(FIPS=analyze.stuff::lead.zeroes(1:1000, 12),
   air=rlnorm(1000), water=rlnorm(1000)*5, stringsAsFactors=FALSE)
 demogdata=data.frame(FIPS=analyze.stuff::lead.zeroes(1:1000, 12),
   pop=rnorm(n=1000, mean=1400, sd=200), stringsAsFactors=FALSE)
 x=ejscreen.lookuptables(envirodata, weights=demogdata$pop, cols='all', zonecols=NULL)

## End(Not run)
```

---

ejscreen.rollup *Summarize EJSCREEN Dataset at Lower Resolution (e.g., Tracts)*

Description

Start with full EJSCREEN dataset at one resolution (typically block groups), and create summary at a higher geographic scale (e.g., tracts or counties)

Usage

```
ejscreen.rollup(bg, fipsname = "FIPS.TRACT", scalename = "tracts", enames,
  folder = getwd(), sumnames, avgnames, wts, ...)
```

Arguments

bg	Data.frame of raw data for environmental and demographic counts, one row per block group typically, one column per indicator.
fipsname	Default is 'FIPS.TRACT' - specifies colname of unique ID field FIPS used to group by. Can be FIPS.TRACT, FIPS.COUNTY, FIPS.ST, or REGION in default dataset.
scalename	Not used. Default is 'tracts' - specifies text to use in naming the saved file.
enames	Default is names.e, the colnames of raw envt indicators in bg
folder	Not used. Optional, default is getwd().
sumnames	Default is a vector of colnames in bg, those which should be rolled up as sums (e.g., sum of all block group population counts in the tract)
avgnames	Default is a vector of colnames in bg, those which should be rolled up as weighted averages (e.g., pop wtd mean of air pollution level)
wts	Default is 'pop', the colname in bg specifying the field to use when calculating the weighted mean of all blockgroups in a tract, for example.
...	Optional parameters to pass to ejscreen.create which uses formulas to create indicators from raw values.
acsnames	Default is a vector of demographic colnames in bg, used in default ejscreen dataset (see code or ejscreenformulas)

Details

**default fieldnames are assumed for now. Uses ejscreen.create

Value

Returns a data.frame with ejscreen dataset of environmental and demographics indicators, and EJ Indexes, as raw values, US percentiles, but not text for popups. Output has one row per tract, county, state, or region, depending on what is specified.

See Also

ejscreen.create

Examples

```
## Not run:
 load("~/Dropbox/EJSCREEN/R analysis/bg 2015-04-22 Rnames plus subgroups.RData")
 # Do this for each of several levels of resolution
 #
fipsnames <- c('FIPS.TRACT', 'FIPS.COUNTY', 'FIPS.ST', 'REGION')
scalenames <- c('tracts', 'counties', 'states', 'regions')
# or just for tracts, say this:
#   fipsnames <- 'FIPS.TRACT'; scalenames <- 'tracts'

for (i in 1:length(fipsnames)) {
```

```
  ####################################
  # Specify resolution of interest
  fipsname <- fipsnames[i] # 'FIPS.TRACT'
  scalename <- scalenames[i] # 'tracts'

  ####################################
  # Get results, using the function
  myrollup <- ejscreen.rollup(bg=bg, fipsname = fipsname, scalename = scalename)

  ####################################
  # Save results
  save(myrollup, file = paste('EJSCREEN 2015', scalename, 'data.RData') )
  write.csv(myrollup, row.names = FALSE, file = paste('EJSCREEN 2015', scalename, 'data.csv'))

}


## End(Not run)
```

`ejscreenformulas` *EJSCREEN 2015 Formulas and Fieldnames*

Description

This provides fieldnames and formulas required by the **ejscreen** package. Formulas can be viewed this way: sort(ejscreenformulas$formula)

Usage

```
data('ejscreenformulas')
```

Format

A data.frame:

> str(ejscreenformulas)
'data.frame': 470 obs. of 8 variables:

- $ gdbfieldname : chr NA NA NA NA ...
- $ Rfieldname : chr "ageunder5m" "age5to9m" "age10to14m" "age15to17m" ...
- $ acsfieldname : chr "B01001.003" "B01001.004" "B01001.005" "B01001.006" ...
- $ type : chr "ACS" "ACS" "ACS" "ACS" ...
- $ glossaryfieldname: chr NA NA NA NA ...
- $ formula : chr NA NA NA NA ...
- $ acsfieldnamelong : chr "Under 5 years|SEX BY AGE" "5 to 9 years|SEX BY AGE" "10 to 14 years|SEX BY AGE" "15 to 17 years|SEX BY AGE" ...
- $ universe : chr "Universe: Total population" "Universe: Total population" "Universe: Total population" "Universe: Total population" ...

Source

See related Technical Documentation at http://www.epa.gov/ejscreen

See Also

ejscreenformulasnoej names.evars names.dvars names.ejvars

`ejscreenformulasnoej` *EJSCREEN 2015 Formulas and Fieldnames Excluding EJ Index Formulas*

Description

This provides fieldnames and formulas required by the **ejscreen** package. Formulas can be viewed this way: sort(ejscreenformulas$formula) This excludes the EJ Index formulas for cases where those are to be calculated using code separately.

Usage

```
data('ejscreenformulasnoej')
```

Format

A data.frame:

> str(ejscreenformulas)
'data.frame': 470 obs. of 8 variables:

- $ gdbfieldname : chr NA NA NA NA ...
- $ Rfieldname : chr "ageunder5m" "age5to9m" "age10to14m" "age15to17m" ...
- $ acsfieldname : chr "B01001.003" "B01001.004" "B01001.005" "B01001.006" ...
- $ type : chr "ACS" "ACS" "ACS" "ACS" ...
- $ glossaryfieldname: chr NA NA NA NA ...
- $ formula : chr NA NA NA NA ...
- $ acsfieldnamelong : chr "Under 5 years|SEX BY AGE" "5 to 9 years|SEX BY AGE" "10 to 14 years|SEX BY AGE" "15 to 17 years|SEX BY AGE" ...
- $ universe : chr "Universe: Total population" "Universe: Total population" "Universe: Total population" "Universe: Total population" ...

Source

See related Technical Documentation at http://www.epa.gov/ejscreen

See Also

ejscreenformulas names.evars names.dvars names.ejvars

ejscreensignifarray *Specify Significant Digits for Each Column of EJSCREEN Indicators*

Description

Given a matrix or numeric data.frame, round each column to a specified column-specific number of significant digits. This function provides default values significant digits to use for an EJSCREEN environmental dataset. This is a wrapper for analyze.stuff::signifarray which is a wrapper that applies signif() to a matrix or data.frame.

Usage

```
ejscreensignifarray(dat, digits = "ejscreen")
```

Arguments

dat	Required, matrix or numeric data.frame with the values to be rounded.
digits	Optional, 'ejscreen' by default. Can be a vector as long as the number of columns in dat, where each elements specifies the number of significant digits to retain for numbers in the corresponding column of dat. If 'ejscreen' it specifies using the default settings described below in details, in which case colnames(dat) must be exactly the same (but in any order) as defaultcolnames below.

Details

Sig figs used if digits specified as 'ejscreen' are those stored in data(esigfigs)

Value

Returns dat, but with numbers rounded based on digits parameter.

See Also

esigfigs signifarray signif

Examples

```
ejscreensignifarray(data.frame(a=rnorm(10), b=rnorm(10), c=rnorm(10)), 1:3)
 envirodata <- data.frame(matrix(rnorm(12*10), ncol=12)); data("names.evars"); names(envirodata) <- names.e
 ejscreensignifarray(envirodata)
```

`esigfigs` *How many signif digits to show*

Description

How many sig figs to show in showing environmental indicators in EJSCREEN 2015-2016?

Usage

```
data('esigfigs')
```

Format

A data.frame:

> str(esigfigs)
'data.frame': 12 obs. of 2 variables:
$ sigfigs: num 3 3 2 2 2 3 2 2 2 2 ...
$ evar : chr "pm" "o3" "cancer" "neuro" ...

sigfigs evar
3 pm
3 o3
2 cancer
2 neuro
2 resp
3 dpm
2 pctpre1960
2 traffic.score
2 proximity.npl
2 proximity.rmp
2 proximity.tsdf
2 proximity.npdes

Source

See related Technical Documentation at http://www.epa.gov/ejscreen

See Also

make.popup.e

make.popup.d *Make text to be shown in popups on Demographic data map*

Description

Takes raw values and what percentiles they are at, and presents those as a text field to be used as the text in a popup window on a map

Usage

```
make.popup.d(d, pctile, prefix = "pctile.text.", basenames)
```

Arguments

d	raw demographic values, 0-1 (such as 0.3345 where roughly 33 percent of the local population is under age 5)
pctile	required integers 0 to 100, representing the percentile(s) at which the raw value(s) fall(s).
prefix	optional, default is 'pctile.text.' This is a text string specifying the first part of the desired resulting fieldname in outputs.
basenames	optional, default is colnames(d). Defines colname(s) of outputs, which are the prefix plus this.

Details

Note d should be a (vector? or) data.frame of exact demographic percentages from 0 to 1, not 0 to 100 BUT pctile should be INTEGER 0 to 100, NOT 0 to 1! Because that is how EJSCREEN data are stored In EJSCREEN, there are three types of pctile.text fields: E (text varies), D, EJ: 'pctile.text.cancer' "55 lifetime risk per million (91 'pctile.text.pctmin' "13 'pctile.text.EJ.DISPARITY.cancer.eo' "36

Value

Returns character vector or data.frame, same shape as first input parameter.

See Also

make.popup.d make.popup.e make.popup.ej pctileAsText

Examples

```
# inputs are test0 and test1, and desired output is like test2 (except note how prefix is added to each basenar
 test0 <- structure(list(
   VSI.eo = c(0.185525372063833, 0.174428104575163, 0.485647788983707),
   pctmin = c(0.131656804733727, 0.111928104575163, 0.671062839410395),
   other = c(NA, NA, 0.02)),
   .Names = c("VSI.eo", "pctmin", "other"),
   row.names = c(NA, 3L), class = "data.frame")
  test0
 # VSI.eo    pctmin other
 # 1 0.1855254 0.1316568    NA
 # 2 0.1744281 0.1119281    NA
```

```
# 3 0.4856478 0.6710628  0.02

 test1 <- structure(list(
   pctile.VSI.eo = c(27.1991395138354, 24.6836238179206, 72.382419748292),
   pctile.pctmin = c(30.2662374847936, 26.761078397073, 78.2620665123235),
   other = c(NA, NA, 4)),
   .Names = c("pctile.VSI.eo", "pctile.pctmin", "other"),
   row.names = c(NA, 3L), class = "data.frame")
 test1
#   pctile.VSI.eo pctile.pctmin other
# 1      27.19914      30.26624    NA
# 2      24.68362      26.76108    NA
# 3      72.38242      78.26207     4

 test2 <- structure(list(
   pctile.text.VSI.eo = c("19% (27%ile)", "17% (24%ile)","49% (72%ile)"),
   pctile.text.pctmin = c("13% (30%ile)", "11% (26%ile)", "67% (78%ile)"),
   other = c(NA, NA, 4)),
   .Names = c("pctile.text.VSI.eo","pctile.text.pctmin", "other"),
   row.names = c(NA, 3L), class = "data.frame")
 test2
#   pctile.text.VSI.eo pctile.text.pctmin other
# 1       19% (27%ile)       13% (30%ile)    NA
# 2       17% (24%ile)       11% (26%ile)    NA
# 3       49% (72%ile)       67% (78%ile)     4

make.popup.d(test0, test1)
 #   pctile.text.VSI.eo pctile.text.pctmin pctile.text.other
 # 1       19% (27%ile)       13% (30%ile)              <NA>
 # 2       17% (24%ile)       11% (26%ile)              <NA>
 # 3       49% (72%ile)       67% (78%ile)        2% (4%ile)
```

`make.popup.e` *Make text to be shown in popups on Envt data map*

Description

Takes raw values and what percentiles they are at, and presents those as a text field to be used as the text in a popup window on a map

Usage

```
make.popup.e(e, pctile, prefix = "pctile.text.", basenames, units, sigfigs)
```

Arguments

`e`	raw environmental indicator values for various locations
`pctile`	required integers 0 to 100, representing the percentile(s) at which the raw value(s) fall(s).
`prefix`	optional, default is 'pctile.text.' This is a text string specifying the first part of the desired resulting fieldname in outputs.
`basenames`	optional, default is colnames(e). Defines colname(s) of outputs, which are the prefix plus this.

units	optional character vector with one per column of e, default is the units used for the 2016 EJSCREEN environmental indicators, such as 'ppb' and 'ug/m3' – function will try to use units appropriate to basenames, looking in data(popupunits), and use '' (blank) if no match is found.
sigfigs	optional, numeric vector with one per col of e, defining number of significant digits to show in popup, defaulting to rules in EJSCREEN 2016 version, or just 2 for basenames not found in data(esigfigs).

Details

Could edit code to NOT put in the units when value is NA? Could edit code to handle cases like only one row, matrix not df? Could fix to use only one space when no units

EJSCREEN as of 2015 used 85 pctile.text. fields, for popup text, like "pctile.text.EJ.DISPARITY.pm.eo" names(bg2)[grepl('pctile.text', names(bg2))] length(bg2[1, grepl('pctile.text', names(bg2))]) # [1] 85

In EJSCREEN, there are three types of pctile.text fields: E (text varies), D, EJ: 'pctile.text.cancer' "55 lifetime risk per million (91 'pctile.text.pctmin' "13 'pctile.text.EJ.DISPARITY.cancer.eo' "36 For E popups, text includes units: (neuro is no longer used in 2016 version of EJSCREEN)

names.e.pctile[names.e.pctile != 'pctile.neuro'] # [1] "pctile.pm" "pctile.o3" "pctile.cancer" # [4] "pctile.resp" "pctile.dpm" "pctile.pctpre1960" # [7] "pctile.traffic.score" "pctile.proximity.npl" "pctile.proximity.rmp" # [10] "pctile.proximity.tsdf" "pctile.proximity.npdes"

NOTE HOW UNITS ARE PART OF THE POPUP, AND IT USES SPECIAL ROUNDING RULES # #' # Stored in data('popunits') # colnames are evar and units

t(bg2[1, gsub('pctile', 'pctile.text', names.e.pctile[names.e.pctile != 'pctile.neuro'])]) # # pctile.text.pm "10.4 ug/m3 (76 # pctile.text.o3 "42.8 ppb (22 # pctile.text.cancer "55 lifetime risk per million (91 # pctile.text.resp "2.1 (72 # pctile.text.dpm "0.401 ug/m3 (24 # pctile.text.pctpre1960 "0.4 = fraction pre-1960 (68 # pctile.text.traffic.score "23 daily vehicles/meters distance (28 # pctile.text.proximity.npl "0.071 sites/km distance (55 # pctile.text.proximity.rmp "0.085 facilities/km distance (21 # pctile.text.proximity.tsdf "0 facilities/km distance (26 # pctile.text.proximity.npdes "0.25 facilities/km distance (70 # t(bg2[125:126, gsub('pctile', 'pctile.text', names.e.pctile[names.e.pctile != 'pctile.neuro'])]) # 125 126 # pctile.text.pm "8.37 ug/m3 (27 # pctile.text.o3 "41.7 ppb (19 # pctile.text.cancer "36 lifetime risk per million (37 # pctile.text.resp "1.4 (37 # pctile.text.dpm "0.275 ug/m3 (13 # pctile.text.pctpre1960 "0.055 = fraction pre-1960 (27 # pctile.text.traffic.score "1.7 daily vehicles/meters distance (6 # pctile.text.proximity.npl "0.056 sites/km distance (47 # pctile.text.proximity.rmp "0.046 facilities/km distance (7 # pctile.text.proximity.tsdf "0 facilities/km distance (26 # pctile.text.proximity.npdes "0.067 facilities/km distance (16 # # single result, e.g.: "24 #

Value

Returns character vector or data.frame, same shape as first input parameter.

See Also

esigfigs make.popup.d make.popup.e make.popup.ej pctileAsText

Examples

```
#  Example: inputs are test0 and test1, and desired output is like test2 (except note how prefix is added to e

  test0 <- structure(list(
    e1 = c(0.185525372063833, 0.174428104575163, 0.485647788983707),
    e2 = c(0.131656804733727, 0.111928104575163, 0.671062839410395),
```

```
  other = c(NA, NA, 0.02)),
  .Names = c("e1", "e2", "other"),
  row.names = c(NA, 3L), class = "data.frame")
test0

 test1 <- structure(list(
  pctile.e1 = c(27.1991395138354, 24.6836238179206, 72.382419748292),
  pctile.e2 = c(30.2662374847936, 26.761078397073, 78.2620665123235),
  other = c(NA, NA, 4)),
  .Names = c("pctile.e1", "pctile.e2", "other"),
  row.names = c(NA, 3L), class = "data.frame")
 test1

test2 <- structure(list(
  pctile.text.e1 = c("19 (27%ile)", "17 (24%ile)","49 (72%ile)"),
  pctile.text.e2 = c("13 (30%ile)", "11 (26%ile)", "67 (78%ile)"),
  other = c(NA, NA, 4)),
  .Names = c("pctile.text.e1","pctile.text.e2", "other"),
  row.names = c(NA, 3L), class = "data.frame")
test2

make.popup.e(test0, test1)
```

`make.popup.ej` *Make text to be shown in popups on EJ map*

Description

Takes percentiles (unlike make.popup.d or make.popup.e, which need raw values too), and presents those as a text field to be used as the text in a popup window on a map.

Usage

```
make.popup.ej(pctile, prefix = "pctile.text.", basenames)
```

Arguments

`pctile`	required integers 0 to 100
`prefix`	optional, default is 'pctile.text.' This is a text string specifying the first part of the desired resulting fieldname in outputs.
`basenames`	optional, default is 'pctile.xxx' where xxx is colnames(pctile). Defines colname(s) of outputs, which are the prefix plus this.

Details

Note pctile should be a (vector? or) data.frame of percentiles as INTEGER 0 to 100, NOT 0 to 1! Because that is how EJSCREEN data are stored. Might add code to handle cases like only one row, matrix not df, etc? Assume normal EJSCREEN pctile cols here would be like pctile.EJ.DISPARITY.pm.eo and then output popup col would be like pctile.text.EJ.DISPARITY.pm.eo In EJSCREEN, there are three types of pctile.text fields: E (text varies), D, EJ: 'pctile.text.cancer' "55 lifetime risk per million (91 'pctile.text.pctmin' "13 'pctile.text.EJ.DISPARITY.cancer.eo' "36

Value

Returns character vector or data.frame, same shape as pctile.

See Also

make.popup.d make.popup.e make.popup.ej pctileAsText

Examples

```
test1 <- structure(list(
   pctile.EJ.DISPARITY.pm.eo = c(43.1816682334032, 27.4198086017171, 71.7852110581344, NA),
   pctile.EJ.DISPARITY.o3.eo = c(47.1675935028896, 33.9578650432096, 69.7501760334948, NA)),
    .Names = c("pctile.EJ.DISPARITY.pm.eo", "pctile.EJ.DISPARITY.o3.eo"),
    row.names = c(1L, 2L, 3L, 126L), class = "data.frame")
  test1
  #    pctile.EJ.DISPARITY.pm.eo pctile.EJ.DISPARITY.o3.eo
  #1                     43.18167                  47.16759
  #2                     27.41981                  33.95787
  #3                     71.78521                  69.75018
  #126                         NA                        NA

  test2 <- structure(list(
    pctile.text.EJ.DISPARITY.pm.eo = c("43%ile", "27%ile", "71%ile", NA),
    pctile.text.EJ.DISPARITY.o3.eo = c("47%ile", "33%ile", "69%ile", NA)),
    .Names = c("pctile.text.EJ.DISPARITY.pm.eo", "pctile.text.EJ.DISPARITY.o3.eo"),
    row.names = c(1L, 2L, 3L, 126L), class = "data.frame")
  test2
  #    pctile.text.EJ.DISPARITY.pm.eo pctile.text.EJ.DISPARITY.o3.eo
  #1                           43%ile                         47%ile
  #2                           27%ile                         33%ile
  #3                           71%ile                         69%ile
  #126                           <NA>                           <NA>

  make.popup.ej(test1)
  #  pctile.text.EJ.DISPARITY.pm.eo pctile.text.EJ.DISPARITY.o3.eo
  #1                         43%ile                        47%ile
  #2                         27%ile                        33%ile
  #3                         71%ile                        69%ile
  #4                           <NA>                          <NA>
```

---

names.dvars *Fieldnames of demographic columns in ejscreen package data*

Description

This data set provides variables that hold the colnames of demographic fields in data.frames that may be used in the ejscreen package to make it easier to refer to them as a vector, e.g., mydf[, names.e]

Usage

```
data('names.dvars'); names.d
```

Format

A series of variables (each is a character vector of colnames):

- "names.d" (VSI.eo, VSI.svi6, pctmin, pctlowinc, pctlths, pctlingiso, pctunder5, pctover64)
- "names.d.bin"
- "names.d.eo"
- "names.d.eo.bin"
- "names.d.eo.pctile"
- "names.d.pctile"
- "names.d.subgroups"
- "names.d.subgroups.count"
- "names.d.subgroups.pct"
- "names.d.svi6"
- "names.d.svi6.bin"
- "names.d.svi6.pctile" #'
- "Dlist" (this one is like names.d, but as a list, not a vector)

Source

Names developed for this package. No external data source.

See Also

`ejscreenformulas` `names.evars` `names.dvars` `names.ejvars`

---

names.ejvars *Fieldnames of environmental justice indicator columns in ejscreen package data*

Description

This data set provides variables that hold the colnames of environmental indicator fields in data.frames that may be used in the ejscreen package to make it easier to refer to them as a vector, e.g., mydf[, names.ej]

Usage

```
data('names.ejvars')
```

Format

A series of variables (each is a character vector of colnames):

- "names.ej"
- "names.ej.bin"
- "names.ej.burden.eo"
- "names.ej.burden.eo.bin"

- "names.ej.burden.eo.pctile"
- "names.ej.burden.svi6"
- "names.ej.burden.svi6.bin"
- "names.ej.burden.svi6.pctile"
- "names.ej.pct.eo"
- "names.ej.pct.eo.bin"
- "names.ej.pct.eo.pctile"
- "names.ej.pct.svi6"
- "names.ej.pct.svi6.bin"
- "names.ej.pct.svi6.pctile"
- "names.ej.pctile"
- "names.ej.svi6"
- "names.ej.svi6.bin"
- "names.ej.svi6.pctile"
- "namesall.ej"
- "namesall.ej.bin"
- "namesall.ej.pctile"

Source

Names developed for this package. No external data source.

See Also

`ejscreenformulas` `names.evars` `names.dvars` `names.ejvars`

`names.evars` *Fieldnames of environmental indicator columns in ejscreen package data*

Description

This data set provides variables that hold the colnames of environmental indicator fields in data.frames that may be used in the ejscreen package to make it easier to refer to them as a vector, e.g., mydf[, names.e]

Usage

```
data('names.evars')
```

Format

A series of variables (each is a character vector of colnames):

- "names.e" (pm, o3, cancer, neuro, resp, dpm, pctpre1960, traffic.score, proximity.npl, proximity.rmp, proximity.tsdf, proximity.npdes)
- "names.e.bin"
- "names.e.pctile"
- "Elist" (this one is like names.e, but as a list, not a vector)

Source

Names developed for this package. No external data source.

See Also

ejscreenformulas names.evars names.dvars names.ejvars

---

pctileAsText *Utility function in showing a percentile as popup text*

Description

Converts numeric percentiles (0-100) into character (text) that converts 95.3124 to '95

Usage

```
pctileAsText(x)
```

Arguments

x	vector or data.frame of numeric values 0 to 100 (not 0 to 1), representing percentiles from EJSCREEN dataset

Value

Returns matrix/vector of same shape as x if x was data.frame/vector

Examples

```
## Not run:
 (bg2[ 125:126, c('pctile.pctmin', 'pctile.EJ.DISPARITY.pm.eo') ])
 (bg2[ 125:126, c('pctile.text.pctmin', 'pctile.text.EJ.DISPARITY.pm.eo') ])
 pctileAsText(bg2[ 125:126, c('pctile.pctmin', 'pctile.EJ.DISPARITY.pm.eo') ])

## End(Not run)
```

---

ustotals *Get US Totals and Percentages Overall for EJSCREEN Fields*

Description

This function simply takes a data.frame of EJSCREEN demographic data and returns the total count or overall US percentage for various fields, by using the appropriate denominator (universe) to calculate any given percentage. For example, PCTLOWINC.US equals sum(lowinc) / sum(povknownratio), not sum(lowinc) / sum(pop). This function is hard-coded to use specified field names referring to EJSCREEN variables. This function is not needed to create an EJSCREEN dataset, but is convenient if one wants US summary values.

Usage

```
ustotals(bg)
```

Arguments

bg — Must be a data.frame that has the following colnames:

- pop,
- lowinc,
- mins,
- under5,
- over64,
- lths,
- lingiso,
- pre1960,
- hisp,
- nhwa,
- nhba,
- nhaiana,
- nhaa,
- nhnhpia,
- nhotheralone,
- nhmulti,
- povknownratio,
- age25up,
- hhlds,
- builtunits

Value

Returns a named list of US totals and percentages (as fractions 0-100) (e.g., POP.US=xxxx, etc.):

- POP.US,
- LOWINC.US,
- MINS.US,
- UNDER5.US,
- OVER64.US,
- LTHS.US,
- LINGISO.US,
- PRE1960.US,
- HISP.US,
- NHWA.US,
- NHBA.US,
- NHAIANA.US,
- NHAA.US,
- NHNHPIA.US,
- NHOTHERALONE.US,
- NHMULTI.US,

- PCTLOWINC.US,
- PCTMIN.US,
- PCTUNDER5.US,
- PCTOVER64.US,
- PCTLTHS.US,
- PCTLINGISO.US,
- PCTPRE1960.US,
- PCTHISP.US,
- PCTNHWA.US,
- PCTNHBA.US,
- PCTNHAIANA.US,
- PCTNHAA.US,
- PCTNHNHPIA.US,
- PCTNHOTHERALONE.US,
- PCTNHMULTI.US

Examples

```
usapprox=data.frame(pop=rep(1419.767,217739),lowinc=464.4692,mins=515.4554,under5=92.48634,
 over64=186.7899,lths=134.0128,lingiso=24.68058, pre1960=183.3237,hisp=232.1370,
 nhwa=904.3119,nhba=173.5408,nhaiana=9.418460, nhaa=67.47893,nhnhpia=2.204764,
 nhotheralone=2.829952,nhmulti=27.84555, povknownratio=1383.92,age25up=938.4447,
 hhlds=529.1969,builtunits=604.5883)
cbind( ustotals(usapprox) )
```

Index