Package 'rFIA'

August 14, 2019

Type Package

Title Spatio-Temporal Estimation of Forest Variables using the USFS Forest Inventory and Analysis Database

Version 0.1.0

Author Hunter Stanke [aut, cre], Andrew Finley [aut]

Maintainer Hunter Stanke <stankehunter5@gmail.com>

Description The goal of rFIA is to increase the accessibility and use of the USFS Forest Inventory and Analysis (FIA) Database by providing a user-friendly, open source toolkit to easily query and analyze FIA Data. Designed to accommodate a wide range of potential user objectives, `rFIA` simplifies the estimation of forest variables from the FIA Database and allows all R users (experts and newcomers alike) to unlock the flexibility and potential inherent to the Enhanced FIA design. Specifically, `rFIA` improves accessibility to the spatio-temporal estimation capacity of the FIA Database by producing space-time indexed summaries of forest variables within user-defined population boundaries. Direct integration with other popular R packages (e.g., dplyr, sp, and sf) facilitates efficient space-time query and data summary, and supports common data representations and API design. The package implements design-based estimation procedures outlined by Bechtold & Patterson (2005), and has been validated against estimates and sampling errors produced by EVALIDator. Current development is focused on the implementation of spatially-enabled model-assisted estimators to improve population, change, and ratio estimates.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.1.0)

Imports dplyr, sp, sf, stringr, forcats, tidyr, parallel, pbapply, ggplot2, gganimate, methods, data.table

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

R topics documented:

`animateFIA` *Animated plots of spatial FIA summaries*

Description

Produce temporally animated plots of spatial FIA summaries. Requires that summaries be returned as spatial objects (ex. specify `returnSpatial = TRUE` when computing summaries using `tpa`), and that multiple reporting years be present in the summary data (animations iterate through time). Specify `savePath` & `fileName` to save animation as .gif file.

Usage

```
animateFIA(data, fillVar, title = NULL, colOption = 'viridis',
           minYear = 2005, direction = 1, alpha = .9,
           transform = "identity", text.size = 1, text.font = '',
           lab.width = 1, legend.height = 1, legend.width = 1,
           savePath, fileName, ...)
```

Arguments

`data`	sf object; spatial FIA summary produced from other `rFIA` functions (ex. `tpa`, `biomass`, etc.).
`fillVar`	character; variable contained in `data` which will be used to fill polygons. Variable name must be quoted.
`title`	character; plot title.

colOption	character; One of: "viridis" (default), "magma", "inferno", "plasma", or "cividis".
minYear	numeric; earliest year to be included in animation. FIA data is sparse in years prior to 2005 and estimates are unlikely to be available.
direction	numeric; sets the order of colors in the scale. If 1, the default, colors are ordered from darkest to lightest. If -1, the order of colors is reversed.
alpha	numeric; alpha transparency, a number in [0,1], see argument alpha in hsv.
transform	character; transformations to apply to plotted variable fillVar. Options include: "asn", "atanh", "boxcox", "exp", "identity", "log", "log10", "log1p", "log2", "logit", "reciprocal", "reverse", "sqrt".
text.size	numeric; scalar for text size (ex. text.size = 2 would be twice the default size).
text.font	character; font family. Choose from: 'Short', 'Canonical', 'mono', 'Courier', 'sans', 'Helvetica', 'serif', 'Times', 'AvantGarde', 'Bookman', 'Helvetica-Narrow', 'NewCenturySchoolbook', 'Palatino', 'URWGothic', 'URWBookman', 'NimbusMon', 'URWHelvetica', 'NimbusSan', 'NimbusSanCond', 'CenturySch', 'URWPalladio', 'URWTimes', or 'NimbusRom'.
lab.width	numeric; scalar for legend title width. This value controls text wrapping in title.
legend.height	numeric; scalar for legend height.
legend.width	numeric; scalar for legend width.
savePath	character; path to save plot to (combined with fileName).
fileName	character; file name to create on disk.
...	Other arguments passed on to scale_fill_viridis_c.

Author(s)

Hunter Stanke & Andrew Finley, Michigan State University

Examples

```
## Not run:
## Compute abundance estimates for live stems in Rhode Island
## for all available inventory years, summarized by counties
tpaRI <- tpa(fiaRI, polys = countiesRI, returnSpatial = TRUE)

## Produce animated plot
animateFIA(tpaRI, fillVar = 'TPA', title = 'Abundance (TPA)')

## With a square root transform
animateFIA(tpaRI, fillVar = 'TPA', title = 'Abundance (TPA)', transform = 'sqrt')

## End(Not run)
```

biomass *Estimate volume, biomass, and carbon stocks from the FIADB*

Description

Produces estimates of volume, biomass, and carbon on a per acre basis from FIA data, along with population estimates for each variable. Estimates can be produced for regions defined within the FIA Database (e.g. counties), at the plot level, or within user-defined areal units. Options to group estimates by species, size class, and other variables defined in the FIADB. If multiple reporting years (EVALIDs) are included in the data, estimates will be output as a time series. If multiple states are represented by the data, estimates will be output for the full region (all area combined), unless specified otherwise (e.g. `grpBy = 'STATECD'`).

Usage

```
biomass(db, grpBy = NULL, polys = NULL, returnSpatial = FALSE, bySpecies = FALSE,
        bySizeClass = FALSE, landType = 'forest', treeType = 'live',
        treeDomain = NULL, areaDomain = NULL, totals = FALSE, byPlot = FALSE,
        SE = TRUE, progress = TRUE, nCores = 1)
```

Arguments

db	`FIA.Database` object produced from `readFIA`; Function requires that PLOT, TREE, COND, POP_PLOT_STRATUM_ASSGN, POP_ESTN_UNIT, POP_EVAL, POP_STRATUM, POP_EVAL_TYP, POP_EVAL_GRP tables exist in FIA.Database object.
grpBy	character; variables from PLOT, COND, or TREE tables to group estimates by. Multiple grouping variables should be combined with `c()`, and grouping will occur heirarchically. For example, to produce seperate estimates for each ownership group within ecoregion subsections, specify `c('ECOSUBCD','OWNGRPCD')`.
polys	`sp` or `sf` Polygon/MultiPolgyon object; Areal units to bin data for estimation. Seperate estimates will be produces for region encompassed by each areal unit.
returnSpatial	logical; if TRUE, return sf spatial object (polys must also be specified).
bySpecies	logical; if TRUE, returns estimates grouped by species.
bySizeClass	logical; if TRUE, returns estimates grouped by size class (2-inch intervals, see `makeClasses` to compute different size class intervals).
landType	character ('forest' or 'timber'); Type of land which estimates will be produced for. Timberland is a subset of forestland (default) which has high site potential and non-reserve status (see details).
treeType	character ('all', 'live', 'dead', or 'gs'); Type of tree which estimates will be produced for. All (default) includes all stems, live and dead, greater than 1 in. DBH. Live/Dead includes all stems greater than 1 in. DBH which are live or dead (leaning less than 45 degrees), respectively. GS (growing-stock) includes live stems greater than 5 in. DBH which contain at least one 8 ft merchantable log.

treeDomain	logical predicates defined in terms of the variables in PLOT, TREE, and/or COND tables. Used to define the type of trees for which estimates will be produced (e.g. DBH greater than 20 inches: `DIA > 20`, Dominant/Co-dominant crowns only: `CCLCD %in% c(2,3)`). Multiple conditions are combined with `&` (and) or `|` (or). Only trees where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
areaDomain	logical predicates defined in terms of the variables in PLOT and/or COND tables. Used to define the area for which estimates will be produced (e.g. within 1 mile of improved road: `RDDISTCD %in% c(1:6)`, Hard maple/basswood forest type: `FORTYPCD == 805`). Multiple conditions are combined with `&` (and) or `|` (or). Only plots within areas where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
totals	logical; if TRUE, return population estimates (e.g. total area, total biomass) along with ratio estimates (e.g. mean biomass per acre).
byPlot	logical; if TRUE, returns estimates for individual plot locations (population totals not returned).
SE	logical; if TRUE, returns estimates with samping error (approx. 5x faster without returning samping errors)
progress	logical; if TRUE, a dynamic progress bar is displayed (approx. 30% faster when FALSE.
nCores	numeric; number of cores to use for parallel implementation. Check available cores using `detectCores`. Default = 1, serial processing.

Details

Estimation of attributes follows the procedures documented in Bechtold and Patterson (2005). Specifically, volume, biomass, and carbon mass per acre are computed using a sample-based ratio-of-means estimator of total volume (carbon or biomass) / total land area within the domain of interest.

Net volume estimates (`NETVOL`) include only the volume of wood in the central stem of a sample tree, from a 1-foot stump to a minimum 4-inch top diameter, or to where the central stem breaks into limbs all of which are greater than 4.0 inches in diameter. Does not include rotten, missing, and form cull portions of the main stem. Saw volume estimates (`SAWVOL`) incldue the net volume in the sawlog portion of the tree, from a 1-foot stump to a 9 inches (hardwood) or 7 inches (softwood) top. All volume estimates are reported in cubic feet (cu. ft. / acre). For estimates in board feet, multiply output values by 12.

Biomass (`BIO`) and carbon (`CARB`) estimates are computed seperately for aboveground (AG) and belowground (BG) stocks, and their totals are the summation of above and belowground stocks. All biomass and carbon estimates are reported in oven-dry mass (short tons). Belowground mass for an individual tree includes modeled estimates for coarse roots (> 0.1"). Above ground mass includes all portions of a tree above the root collar, excluding foliage.

Stratified random sampling techniques are most often employed to compute estimates in recent inventories, although double sampling and simple random sampling may be employed for early inventories. Estimates are adjusted for non-response bias by assuming attributes of non-response plot locations to be equal to the mean of other plots included within thier respective stratum or population.

Forest land must be at least 10-percent stocked by trees of any size, including land that formerly had such tree cover and that will be naturally or artificially regenerated. Forest land includes transition zones, such as areas between heavily forested and nonforested lands that are at least 10-percent stocked with trees and forest areas adjacent to urban and builtup lands. The minimum area for classification of forest land is 1 acre and 120 feet wide measured stem-to-stem from the outermost edge. Unimproved roads and trails, streams, and clearings in forest areas are classified as forest if less than 120 feet wide. Timber land is a subset of forest land that is producing or is capable of producing crops of industrial wood and not withdrawn from timber utilization by statute or administrative regulation. (Note: Areas qualifying as timberland are capable of producing at least 20 cubic feet per acre per year of industrial wood in natural stands. Currently inaccessible and inoperable areas are NOT included).

Easy, efficient parallelization is implemented with the `parallel` package. Users must only specify the `nCores` argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (`nCores = 1`).

Value

Dataframe or SF object (if `returnSpatial = TRUE`). If `byPlot = TRUE`, totals are returned for each plot. All variables with names ending in `SE` represent the estimate of sampling error (%) of the variable. All variables with names ending in `TOTAL` represent the population total of the variable.

- **YEAR**: reporting year associated with estimates
- **NETVOL_ACRE**: estimate of mean net volume per acre (cu.ft./acre)
- **SAWVOL_ACRE**: estimate of mean merchantable saw volume per acre (cu.ft./acre)
- **BIO_AG_ACRE**: estimate of mean aboveground biomass per acre (tons/acre)
- **BIO_BG_ACRE**: estimate of mean belowground biomass per acre (tons/acre)
- **BIO_ACRE**: estimate of mean total biomass per acre (tons/acre)
- **CARB_AG_ACRE**: estimate of mean aboveground carbon per acre (tons/acre)
- **CARB_BG_ACRE**: estimate of mean belowground carbon per acre (tons/acre)
- **CARB_ACRE**: estimate of mean total carbon per acre (tons/acre)
- **nPlots_VOL**: number of non-zero plots used to compute volume, biomass, and carbon estimates
- **nPlots_AREA**: number of non-zero plots used to compute land area estimates

Note

All sampling error estimates are returned as percentages, and represent ~68% confidence (1 standard deviation). To compute sampling error percent with 95% confidence, multiply by 1.96.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

Bechtold, W.A.; Patterson, P.L., eds. 2005. The Enhanced Forest Inventory and Analysis Program - National Sampling Design and Estimation Procedures. Gen. Tech. Rep. SRS - 80. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 85 p. https://www.srs.fs.usda.gov/pubs/gtr/gtr_srs080/gtr_srs080.pdf

See Also

tpa, vitalRates, growMort

Examples

```
## Not run:

## Load data from the rFIA package
library(rFIA)
data(fiaRI)
data(countiesRI)

## Most recent estimates for live stems on forest land by species
biomass(db = clipFIA(fiaRI, mostRecent = TRUE),
        landType = 'forest',
        treeType = 'live',
        bySpecies = TRUE)

## Same as above, but implemented in parallel (much quicker)
detectCores() # 4 cores available, we will take 3
biomass(db = clipFIA(fiaRI, mostRecent = TRUE),
        landType = 'forest',
        treeType = 'live',
        bySpecies = TRUE,
        nCores = 3)

## Most recent estimates for growing-stock on timber land by species
biomass(db = clipFIA(fiaRI, mostRecent = TRUE),
        landType = 'timber',
        treeType = 'gs',
        bySpecies = TRUE)

## Estimates for snags greater than 20 in DBH on forestland for all
##  available inventories (time-series)
biomass(db = fiaRI,
        landType = 'forest',
        treeType = 'dead',
        treeDomain = DIA > 20)

## Most recent estimates grouped by stand age on forest land
# Make a categorical variable which represents stand age (grouped by 10 yr intervals)
fiaRI$COND$STAND_AGE <- makeClasses(fiaRI$COND$STDAGE, interval = 10)
biomass(db = clipFIA(fiaRI, mostRecent = TRUE),
```

```
        grpBy = 'STAND_AGE')

## Estimates for live white pine ( > 12" DBH) on forested mesic sites (all available inventories)
biomass(fiaRI,
        treeType = 'live',
        treeDomain = SPCD == 129 & DIA > 12, # Species code for white pine
        areaDomain = PHYSCLCD %in% 21:29) # Mesic Physiographic classes

## Most recent estimates for all stems on forest land grouped by user-defined areal units
ctSF <- biomass(clipFIA(fiaRI, mostRecent = TRUE),
                polys = countiesRI,
                returnSpatial = TRUE)
plot(ctSF) # Plot multiple variables simultaneously
plotFIA(ctSF, 'BIO_AG_ACRE') # Plot of aboveground biomass per acre


## End(Not run)
```

---

clipFIA *Spatial and temporal queries for FIADB*

Description

Performs space-time queries on Forest Inventory and Analysis Database (FIADB). Subset database to include only data associated with particular inventory years (i.e., most recent), and/or only data within a user-defined region.

Usage

```
clipFIA(db, mostRecent = TRUE, mask = NULL, matchEval = FALSE,
        evalID = NULL, designCD = NULL)
```

Arguments

db	FIA.Database object produced from readFIA.
mostRecent	logical; if TRUE, returns only data for most recent inventory.
mask	sp or sf Polygon/MultiPolgyon object; defines the boundaries of spatial intersection with FIA tables.
matchEval	logical; if TRUE, returns subset of data for which there are matching reporting years across states. Only useful if db contains mulitple state subsets of the FIA database.
evalID	character; unique value which identifies an inventory year and inventory type for a state. If you would like to subset data for an inventory year other than the most recent, use findEVALID to look locate this value (see Examples below).
designCD	character vector; plot designs to include. Default includes standard national plot design with other similar sampling designs. See FIA Database User Guide Appendix 1 for descriptions of plot designs (see References).

Details

Not required to run other **rFIA** functions, but may help conserve free memory and reduce processing time if user is interested in producing estimates for a specific inventory year or within a region not explicitly described in the database (w/in user defined polygons).

Spatial intersections do not adhere strictly to absolute plot locations, all plots which fall within an estimation unit (often a county) which intersects with a user defined region will be returned. The plots which fall slightly outside of the region do NOT bias estimates (removed from computations), but as FIA often employs stratified random sampling estimators, all plots within intersecting estimation units must be present to proudce unbiased variance estimates.

Value

List object containing spatially intersected FIADB tables.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

See Also

findEVALID

Examples

```
## Load data from rFIA package
library(rFIA)
data(fiaRI)

## Most recent inventory
clipFIA(fiaRI, mostRecent = TRUE)

## Only plots w/in estimation units w/in a user defined polygon
clipFIA(fiaRI, mask = countiesRI[1,])
```

---

countiesRI *County boundaries of Rhode Island*

Description

`sp SpatialPolygonsDataFrame` representing county boundaries in the state of Rhode Island. Specify `countiesRI` as the `polys` argument with `fiaRI` as the `db` argument in any `rFIA` function to produce estimates summarized by these areal units within the state of Rhode Island.

Usage

```
data("countiesRI")
```

Format

Formal class SpatialPolygonsDataFrame

Examples

```
data(countiesRI)
```

`diversity` *Estimate species diversity from FIADB*

Description

Produces estimates of species diversity from FIA data. Returns shannon's index, shannon's equitability, and species richness for alpha (mean/SE of stands), beta, and gamma diversity. Estimates can be produced for regions defined within the FIA Database (e.g. counties), at the plot level, or within user-defined areal units. Options to group estimates by size class and other variables defined in the FIADB. If multiple reporting years (EVALIDs) are included in the data, estimates will be output as a time series. If multiple states are represented by the data, estimates will be output for the full region (all area combined), unless specified otherwise (e.g. `grpBy = 'STATECD'`). Easy options to implement parallel processing.

Usage

```
diversity(db, grpBy = NULL, polys = NULL, returnSpatial = FALSE, bySizeClass = FALSE,
          landType = 'forest', treeType = 'live', treeDomain = NULL,
        areaDomain = NULL, byPlot = FALSE, SE = TRUE, progress = TRUE, nCores = 1)
```

Arguments

`db`	`FIA.Database` object produced from `readFIA`. Function requires that PLOT, TREE, COND, POP_PLOT_STRATUM_ASSGN, POP_ESTN_UNIT, POP_EVAL, POP_STRATUM, POP_EVAL_TYP, POP_EVAL_GRP tables exist in FIA.Database object.
`grpBy`	character; variables from PLOT, COND, or TREE tables to group estimates by. Multiple grouping variables should be combined with `c()`, and grouping will occur heirarchically. For example, to produce seperate estimates for each ownership group within ecoregion subsections, specify `c('ECOSUBCD', 'OWNGRPCD')`.
`polys`	`sp` or `sf` Polygon/MultiPolgyon object; Areal units to bin data for estimation. Seperate estimates will be produces for region encompassed by each areal unit.
`returnSpatial`	logical; if TRUE, return sf spatial object (polys must also be specified).
`bySizeClass`	logical; if TRUE, returns estimates grouped by size class (default 2-inch intervals, see `makeClasses` to compute other size class intervals).

landType	character ('forest' or 'timber'); Type of land which estimates will be produced for. Timberland is a subset of forestland (default) which has high site potential and non-reserve status (see details).
treeType	character ('all', 'live', 'dead', or 'gs'); Type of tree which estimates will be produced for. All (default) includes all stems, live and dead, greater than 1 in. DBH. Live/Dead includes all stems greater than 1 in. DBH which are live or dead (leaning less than 45 degrees), respectively. GS (growing-stock) includes live stems greater than 5 in. DBH which contain at least one 8 ft merchantable log.
treeDomain	logical predicates defined in terms of the variables in PLOT, TREE, and/or COND tables. Used to define the type of trees for which estimates will be produced (e.g. DBH greater than 20 inches: `DIA > 20`, Dominant/Co-dominant crowns only: `CCLCD %in% c(2,3))`. Multiple conditions are combined with `&` (and) or `|` (or). Only trees where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
areaDomain	logical predicates defined in terms of the variables in PLOT and/or COND tables. Used to define the area for which estimates will be produced (e.g. within 1 mile of improved road: `RDDISTCD %in% c(1:6)`, Hard maple/basswood forest type: `FORTYPCD == 805)`. Multiple conditions are combined with `&` (and) or `|` (or). Only plots within areas where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
byPlot	logical; if TRUE, returns estimates for individual plot locations (population totals not returned).
SE	logical; if TRUE, returns estimates with samping error (approx. 5x faster without returning samping errors)
progress	logical; if TRUE, a dynamic progress bar is displayed (approx. 30% faster when FALSE.
nCores	numeric; number of cores to use for parallel implementation. Check available cores using `detectCores`. Default = 1, serial processing.

Details

Procedures for computing diversity indices are outlined in Hill (1973) and Shannon (1948), and estimation of mean/ sampling error follows the procedures documented in Bechtold and Patterson (2005).

Alpha-level indices are computed as the mean (SE) species diversity of a stand or condition as defined by FIA. Specifically, alpha diversity is estimated using a sample-based ratio-of-means estimator of stand diversity (e.g. Richness) * land area of stand / total land area within the domain of interest. Thus estimates of alpha diversity within a stand are weighted by the area which that stand represents. Gamma-level diversity is computed as regional indicies, pooling all plot data. Beta diversity is computed as gamma diversity - alpha diversity, and thus represents the excess of regional diversity with respect to local diversity. As stems of various size classes do not have equal probability of detection under the National FIA plot design, we compute relative proportions of species abundances based on the trees per acre they represent.

Stratified random sampling techniques are most often employed to compute estimates in recent inventories, although double sampling and simple random sampling may be employed for early

inventories. Estimates are adjusted for non-response bias by assuming attributes of non-response plot locations to be equal to the mean of other plots included within thier respective stratum or population.

Forest land must be at least 10-percent stocked by trees of any size, including land that formerly had such tree cover and that will be naturally or artificially regenerated. Forest land includes transition zones, such as areas between heavily forested and nonforested lands that are at least 10-percent stocked with trees and forest areas adjacent to urban and builtup lands. The minimum area for classification of forest land is 1 acre and 120 feet wide measured stem-to-stem from the outermost edge. Unimproved roads and trails, streams, and clearings in forest areas are classified as forest if less than 120 feet wide. Timber land is a subset of forest land that is producing or is capable of producing crops of industrial wood and not withdrawn from timber utilization by statute or administrative regulation. (Note: Areas qualifying as timberland are capable of producing at least 20 cubic feet per acre per year of industrial wood in natural stands. Currently inaccessible and inoperable areas are NOT included).

Easy, efficient parallelization is implemented with the `parallel` package. Users must only specify the `nCores` argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (`nCores = 1`).

Value

Dataframe or SF object (if `returnSpatial = TRUE`). If `byPlot = TRUE`, indices are returned for each plot. All variables with names ending in `SE`, represent the estimate of sampling error (%) of the variable.

- **H_a**: mean Shannon's Diversity Index, alpha (stand) level
- **H_b**: Shannon's Diversity Index, beta (landscape) level
- **H_g**: Shannon's Diversity Index, gamma (regional) level
- **Eh_a**: mean Shannon's Equitability Index, alpha (stand) level
- **Eh_b**: Shannon's Equitability Index, beta (landscape) level
- **Eh_g**: Shannon's Equitability Index, alpha (stand) level
- **S_a**: mean Species Richness, alpha (stand) level
- **S_b**: Species Richness, beta (landscape) level
- **S_g**: Species Richness, gamma (regional) level
- **nStands**: number of stands with non-zero plots used to compute alpha diversity estimates

Note

All sampling error estimates are returned as percentages, and represent ~68% confidence (1 standard deviation). To compute sampling error percent with 95% confidence, multiply by 1.96.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

Bechtold, W.A.; Patterson, P.L., eds. 2005. The Enhanced Forest Inventory and Analysis Program - National Sampling Design and Estimation Procedures. Gen. Tech. Rep. SRS - 80. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 85 p. https://www.srs.fs.usda.gov/pubs/gtr/gtr_srs080/gtr_srs080.pdf

Analysis of ecological communities. (2002). United States: M G M SOFTWARE DESIGN (OR).

Hill, M. O. (1973). Diversity and Evenness: A Unifying Notation and Its Consequences. Ecology, 54(2), 427-432. doi:10.2307/1934352.

Shannon, C. E. (1948). A Mathematical Theory of Communication. Bell System Technical Journal, 27(3), 379-423. doi:10.1002/j.1538-7305.1948.tb01338.x.

See Also

tpa, standStruct, invasive

Examples

```
## Not run:

## Load data from rFIA package
library(rFIA)
data(fiaRI)
data(countiesRI)


## Most recent estimates for live stems on forest land
diversity(db = clipFIA(fiaRI, mostRecent = TRUE),
         landType = 'forest',
         treeType = 'live')

## Same as above, implemented in parallel
detectCores() # 4 cores available, we will take 3
diversity(db = clipFIA(fiaRI, mostRecent = TRUE),
         landType = 'forest',
         treeType = 'live',
         nCores = 3)

## Most recent estimates for growing-stock on timber land by species
diversity(db = clipFIA(fiaRI, mostRecent = TRUE),
          landType = 'timber',
          treeType = 'gs',
          bySizeClass = TRUE)

## Most recent estimates grouped by stand age on forest land
# Make a categorical variable which represents stand age (grouped by 10 yr intervals)
fiaRI$COND$STAND_AGE <- makeClasses(fiaRI$COND$STDAGE, interval = 10)
diversity(db = clipFIA(fiaRI, mostRecent = TRUE),
          grpBy = 'STAND_AGE')
```

```
## Estimates for live white pine ( > 12" DBH) on forested mesic sites (all available inventories)
diversity(fiaRI,
          treeType = 'live',
          treeDomain = DIA > 12,
          areaDomain = PHYSCLCD %in% 21:29) # Mesic Physiographic classes

## Most recent estimates for all stems on forest land grouped by user-defined areal units
ctSF <- diversity(clipFIA(fiaRI, mostRecent = TRUE),
                  polys = countiesRI,
                  returnSpatial = TRUE)
plot(ctSF) # Plot multiple variables simultaneously
plotFIA(ctSF, 'H_a') # Plot of mean Shannons Index of stands

## End(Not run)
```

---

dwm *Estimate volume, biomass, and carbon stocks of down woody material (fuels) from FIADB*

Description

Produces estimates of down woody material stocks on a per acre basis from the Forest Inventory and Analysis Database (FIADB), along with population totals for each variable. Estimates are returned by fuel class (1HR, 10HR, 100HR, 1000HR, Piles) for application in fuels management. Estimates can be produced for regions defined within the FIA Database (e.g. counties), at the plot level, or within user-defined areal units. If multiple reporting years (EVALIDs) are included in the data, estimates will be output as a time series. If multiple states are represented by the data, estimates will be output for the full region (all area combined), unless specified otherwise (e.g. `grpBy = 'STATECD'`). Easy options to implement parallel processing.

Usage

```
dwm(db, grpBy = NULL, polys = NULL, returnSpatial = FALSE, landType = 'forest',
    areaDomain = NULL, byPlot = FALSE, totals = FALSE, tidy = TRUE,
    SE = TRUE, progress = TRUE, nCores = 1)
```

Arguments

db	`FIA.Database` object produced from `readFIA`. Function requires that PLOT, COND_DWM_CALC, COND, POP_PLOT_STRATUM_ASSGN, POP_ESTN_UNIT, POP_EVAL, POP_STRATUM, POP_EVAL_TYP, POP_EVAL_GRP tables exist in FIA.Database object.
grpBy	character; variables from PLOT, COND, or TREE tables to group estimates by. Multiple grouping variables should be combined with `c()`, and grouping will occur heirarchically. For example, to produce seperate estimates for each ownership group within ecoregion subsections, specify `c('ECOSUBCD','OWNGRPCD')`.
polys	`sp` or `sf` Polygon/MultiPolgyon object; Areal units to bin data for estimation. Seperate estimates will be produces for region encompassed by each areal unit.

returnSpatial	logical; if TRUE, return sf spatial object (polys must also be specified).
landType	character ('forest' or 'timber'); Type of land which estimates will be produced for. Timberland is a subset of forestland (default) which has high site potential and non-reserve status (see details).
areaDomain	Logical predicates defined in terms of the variables in PLOT and/or COND tables. Used to define the area for which estimates will be produced (e.g. within 1 mile of improved road: `RDDISTCD %in% c(1:6)`, Hard maple/basswood forest type: `FORTYPCD == 805`). Multiple conditions are combined with `&` (and) or `|` (or). Only plots within areas where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
byPlot	logical; if TRUE, returns estimates for individual plot locations (population totals not returned).
totals	logical; if TRUE, returns population estimates in addition to ratios.
tidy	logical; if TRUE, returns estimates grouped by fuel type, rather than including individual columns for each fuel type (For use in tidyverse packages, e.g. `ggplot2`, `dplyr`). Not recommended when returning spatial objects (`returnSpatial = TRUE`), for consistency with shapefile data structures.
SE	logical; if TRUE, returns estimates with samping error (approx. 5x faster without returning samping errors)
progress	logical; if TRUE, a dynamic progress bar is displayed (approx. 30% faster when FALSE.
nCores	numeric; number of cores to use for parallel implementation. Check available cores using `detectCores`. Default = 1, serial processing.

Details

Estimation of attributes follows the procedures documented in Bechtold and Patterson (2005). Specifically, per acre estimates are computed using a sample-based ratio-of-means estimator of total volume (biomass or carbon) / total land area within the domain of interest.

As defined by FIA, down woody material includes dead organic materials (resulting from plant mortality and leaf turnover) and fuel complexes of live shrubs and herbs. To maintain relevance for forest fuels management, we report estimates grouped by fuel lag-time classes. Specifically, we report estimates for 1HR fuels (small, fine woody debris), 10HR fuels (medium, fine woody debris), 100HR fuels (large, fine woody debris), 1000HR fuels (coarse woody debris), and slash piles. See Woodall and Monleon (2007) for definitions of fuel lag-time classes and for details on sampling and estimation procedures.

Stratified random sampling techniques are most often employed to compute estimates in recent inventories, although double sampling and simple random sampling may be employed for early inventories. Estimates are adjusted for non-response bias by assuming attributes of non-response plot locations to be equal to the mean of other plots included within thier respective stratum or population.

Forest land must be at least 10-percent stocked by trees of any size, including land that formerly had such tree cover and that will be naturally or artificially regenerated. Forest land includes transition zones, such as areas between heavily forested and nonforested lands that are at least 10-percent stocked with trees and forest areas adjacent to urban and builtup lands. The minimum area for

classification of forest land is 1 acre and 120 feet wide measured stem-to-stem from the outermost edge. Unimproved roads and trails, streams, and clearings in forest areas are classified as forest if less than 120 feet wide. Timber land is a subset of forest land that is producing or is capable of producing crops of industrial wood and not withdrawn from timber utilization by statute or administrative regulation. (Note: Areas qualifying as timberland are capable of producing at least 20 cubic feet per acre per year of industrial wood in natural stands. Currently inaccessible and inoperable areas are NOT included).

Easy, efficient parallelization is implemented with the `parallel` package. Users must only specify the `nCores` argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (`nCores = 1`).

Value

Dataframe or SF object (if `returnSpatial = TRUE`). If `byPlot = TRUE`, values are returned for each plot. All variables with names ending in `SE`, represent the estimate of sampling error (%) of the variable. All variables with names ending in `TOTAL` represent the population total of the variable.

- **YEAR**: reporting year associated with estimates
- **VOL_ACRE**: estimate of mean volume per acre of dwm (cu.ft/acre)
- **BIO_ACRE**: estimate of mean biomass per acre of dwm (tons/acre)
- **CARB_ACRE**: estimate of mean carbon mass per acre of dwm (tons/acre)
- **nPlots**: number of non-zero plots used to compute estimates

Note

All sampling error estimates are returned as percentages, and represent ~68% confidence (1 standard deviation). To compute sampling error percent with 95% confidence, multiply by 1.96.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

Bechtold, W.A.; Patterson, P.L., eds. 2005. The Enhanced Forest Inventory and Analysis Program - National Sampling Design and Estimation Procedures. Gen. Tech. Rep. SRS - 80. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 85 p. https://www.srs.fs.usda.gov/pubs/gtr/gtr_srs080/gtr_srs080.pdf

Woodall, C.; Monleon, V.J., eds. 2007. Sampling Protocol, Estimation, and Analysis Procedures for the Down Woody Materials Indicator of the FIA Program. Gen. Tech. Rep. NRS - 22. ewtown Square, PA: U.S. Department of Agriculture, Forest Service, Northern Research Station. https://www.nrs.fs.fed.us/pubs/gtr/gtr_nrs22.pdf

See Also

tpa, biomass

Examples

```
## Not run:

## Load data from rFIA package
library(rFIA)
data(fiaRI)
data(countiesRI)

## Estimates of all forestland, over time
dwm(fiaRI)

## Same as above, but output contains seperate column for each structural stage,
##   rathern than grouping variable
dwm(fiaRI,
    tidy = FALSE)

## Same as above, but implemented in parallel (much quicker)
detectCores() # 4 cores available, we will take 3
dwm(fiaRI,
    tidy = FALSE,
    nCores = 3)

## Estimates of all forestland by owner group (most recent subset)
dwm(clipFIA(fiaRI, mostRecent = TRUE),
    grpBy = c('OWNGRPCD'))

## Estimates of all forestland on mesic sites (most recent)
dwm(clipFIA(fiaRI, mostRecent = TRUE),
    areaDomain = PHYSCLCD %in% 21:29)

## Estimates of all forestland by county and return
#   return spatial object
dwmSF <- dwm(clipFIA(fiaRI, mostRecent = TRUE),
            polys = countiesRI,
            returnSpatial = TRUE,
            tidy = FALSE)
plot(dwmSF)
plotFIA(dwmSF, 'BIO_ACRE') # TOTAL BIOMASS / ACRE (tons)

## End(Not run)
```

Description

Subset of the Forest Inventory and Analysis Database for the state of Rhode Island. Reporting years range from 1985 - 2017. Specify `fiaRI` as the `db` argument in any `rFIA` function to produce estimates for the state fo Rhode Island.

Download other subsets of the FIA Database from the FIA Datamart: https://apps.fs.usda.gov/fia/datamart/datamart.html. Once downloaded, unzip the directory, and read into R using `readFIA`.

Usage

```
data("fiaRI")
```

Format

—- FIA Database Object —-

Reporting Years: 1985 1998 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017

States: RHODE ISLAND

Total Plots: 222

Memory Used: 35.5 Mb

Tables: COND_DWM_CALC COND INVASIVE_SUBPLOT_SPP PLOT POP_ESTN_UNIT POP_EVAL_GRP POP_EVAL_TYP POP_EVAL POP_PLOT_STRATUM_ASSGN POP_STRATUM SUBP_COND_CHNG_MTRX SUBPLOT TREE_GRM_COMPONENT TREE_GRM_ESTN TREE

Examples

```
data(fiaRI)
summary(fiaRI)
print(fiaRI)
```

---

findEVALID *Find EVALIDs used in the FIADB*

Description

Lookup Evaluation IDs (EVALIDs) associated with reporting years and evaluation types used in the Forest Inventory and Analysis Database. NOT required to run other **rFIA** functions. Only use if you are interested in subsetting an `FIA.Database` object for a specific reporting year or evaluation type using `clipFIA`.

Usage

```
findEVALID(db, mostRecent = TRUE, state = NULL, year = NULL, type = NULL)
```

Arguments

db	list; FIA Database object produced from readFIA.
mostRecent	logical; if TRUE, returns EVALIDs associated with most recent inventory.
state	character vector containing full names of states of interest (e.g. c('Michigan','Minnesota','Wisconsi
year	numeric vector containing years of interest (e.g. c(2015,2016,2017))
type	character ('ALL', 'CURR', 'VOL', 'GROW', 'MORT', 'REMV', 'CHANGE', 'DWM', 'REGEN'). See Reference Population Evaluation Table Description Type Table in FIADB P2 User Guide (References) for descriptions of evaluation types.

Details

EVALIDs in the FIA Database are used to reference data points associated with particular inventory years and evaluation types within a state (e.g. 2017 Current Volume in Michigan). They are often extraordinarily confusing for those not familiar for the FIA Database. With this in mind, **rFIA** has been designed to eliminate users dependence on identifying and specifying appropriate EVALIDs to produce desired estimates, and we therefore do not recommend users attempt to identify EVALIDs independently.

Any state or year specified must be present in db to return associated EVALIDS.

Value

A numeric vector containing the EVALIDs associated with states, years, or evaluation types specified.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

See Also

clipFIA

Examples

```
library(rFIA)

## Lookup all EVALIDs in an FIA.Database object
findEVALID(fiaRI, mostRecent = FALSE)

## Find the most recent EVALIDs
findEVALID(fiaRI)
```

growMort *Estimate recruitment, mortality, and harvest rates from FIADB*

Description

Produces estimates of annual regeneration, recruitment, natural mortality, and harvest rates from the Forest Inventory and Analysis Database (FIADB), along with population estimates for each variable. Estimates can be produced for regions defined within the FIA Database (e.g. counties), at the plot level, or within user-defined areal units. Options to group estimates by species, size class, and other variables defined in the FIADB. If multiple reporting years (EVALIDs) are included in the data, estimates will be output as a time series. If multiple states are represented by the data, estimates will be output for the full region (all area combined), unless specified otherwise (e.g. `grpBy = 'STATECD'`). Easy options to implement parallel processing.

Usage

```
growMort(db, grpBy = NULL, polys = NULL, returnSpatial = FALSE, bySpecies = FALSE,
         bySizeClass = FALSE, landType = 'forest', treeType = 'all',
         treeDomain = NULL, areaDomain = NULL, totals = FALSE, byPlot = FALSE,
         SE = TRUE, progress = TRUE, nCores = 1)
```

Arguments

db	`FIA.Database` object produced from `readFIA`. Function requires that PLOT, TREE, TREE_GRM_COMPONENT, and TREE_GRM_ESTN, COND, POP_PLOT_STRATUM_ASSG POP_ESTN_UNIT, POP_EVAL, POP_STRATUM, POP_EVAL_TYP, POP_EVAL_GRP, and SUBP_COND_CHNG_MTRX tables exist in FIA.Database object.
grpBy	character; variables from PLOT, COND, or TREE tables to group estimates by. Multiple grouping variables should be combined with `c()`, and grouping will occur heirarchically. For example, to produce seperate estimates for each ownership group within ecoregion subsections, specify `c('ECOSUBCD','OWNGRPCD')`.
polys	`sp` or `sf` Polygon/MultiPolgyon object; Areal units to bin data for estimation. Seperate estimates will be produces for region encompassed by each areal unit.
returnSpatial	logical; if TRUE, return sf spatial object (polys must also be specified).
bySpecies	logical; if TRUE, returns estimates grouped by species.
bySizeClass	logical; if TRUE, returns estimates grouped by size class (2-inch intervals, see `makeClasses` to compute different size class intervals).
landType	character ('forest' or 'timber'); Type of land which estimates will be produced for. Timberland is a subset of forestland (default) which has high site potential and non-reserve status (see details).
treeType	character ('all', 'live', 'dead', or 'gs'); Type of tree which estimates will be produced for. All (default) includes all stems, live and dead, greater than 1 in. DBH. Live/Dead includes all stems greater than 1 in. DBH which are live or dead (leaning less than 45 degrees), respectively. GS (growing-stock) includes live stems greater than 5 in. DBH which contain at least one 8 ft merchantable log.

treeDomain	logical predicates defined in terms of the variables in PLOT, TREE, and/or COND tables. Used to define the type of trees for which estimates will be produced (e.g. DBH greater than 20 inches: `DIA > 20`, Dominant/Co-dominant crowns only: `CCLCD %in% c(2,3)`). Multiple conditions are combined with `&` (and) or `|` (or). Only trees where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
areaDomain	logical predicates defined in terms of the variables in PLOT and/or COND tables. Used to define the area for which estimates will be produced (e.g. within 1 mile of improved road: `RDDISTCD %in% c(1:6)`, Hard maple/basswood forest type: `FORTYPCD == 805`). Multiple conditions are combined with `&` (and) or `|` (or). Only plots within areas where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
totals	logical; if TRUE, return population estimates (e.g. total area, total mortality) along with ratio estimates (e.g. mean mortality trees per acre).
byPlot	logical; if TRUE, returns estimates for individual plot locations (population totals not returned).
SE	logical; if TRUE, returns estimates with samping error (approx. 5x faster without returning samping errors)
progress	logical; if TRUE, a dynamic progress bar is displayed (approx. 30% faster when FALSE.
nCores	numeric; number of cores to use for parallel implementation. Check available cores using `detectCores`. Default = 1, serial processing.

Details

Estimation of attributes follows the procedures documented in Bechtold and Patterson (2005). Average annual rates are computed using a sample-based ratio of means estimator of total trees subject to an event (e.g. recruitment, mortality) annually / total area. Similarly, the proportion of individuals subject to each event annually is computed as the total trees subject to the event / total trees. All estimates are returned as average annual rates. Only conditions which were forested in time 1 and in time 2 are included in estimates (excluding converted stands).

Recruitment events are defined as when a live stem which is less than 5 inches DBH at time 1, grows to or beyond 5 inches DBH by time 2. This does NOT include stems which grow beyond the 5-inch diameter criteria and are then subject to mortality prior to remeasurement. Natural mortality is defined as when a live stem is subject to non-harvest mortality between successive measurement periods. Finally, harvest is defined as when a live stem is cut and removed between successive measurements.

Stratified random sampling techniques are most often employed to compute estimates in recent inventories, although double sampling and simple random sampling may be employed for early inventories. Estimates are adjusted for non-response bias by assuming attributes of non-response plot locations to be equal to the mean of other plots included within thier respective stratum or population.

Forest land must be at least 10-percent stocked by trees of any size, including land that formerly had such tree cover and that will be naturally or artificially regenerated. Forest land includes transition zones, such as areas between heavily forested and nonforested lands that are at least 10-percent stocked with trees and forest areas adjacent to urban and builtup lands. The minimum area for

classification of forest land is 1 acre and 120 feet wide measured stem-to-stem from the outermost edge. Unimproved roads and trails, streams, and clearings in forest areas are classified as forest if less than 120 feet wide. Timber land is a subset of forest land that is producing or is capable of producing crops of industrial wood and not withdrawn from timber utilization by statute or administrative regulation. (Note: Areas qualifying as timberland are capable of producing at least 20 cubic feet per acre per year of industrial wood in natural stands. Currently inaccessible and inoperable areas are NOT included).

Easy, efficient parallelization is implemented with the `parallel` package. Users must only specify the `nCores` argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (`nCores = 1`).

Value

Dataframe or SF object (if `returnSpatial = TRUE`). If `byPlot = TRUE`, values are returned for each plot. All variables with names ending in `SE`, represent the estimate of sampling error (%) of the variable. All variables with names ending in `TOTAL` represent the population total of the variable.

- **YEAR**: reporting year associated with estimates
- **RECR_TPA**: estimate of mean annual recruitment as trees per acre
- **MORT_TPA**: estimate of mean annual mortality as trees per acre
- **REMV_TPA**: estimate of mean annual removals (harvest) as trees per acre
- **RECR_PERC**: estimate of mean percent of individuals subject to recruitment annually
- **MORT_PERC**: estimate of mean percent of individuals subject to mortality annually
- **REMV_PERC**: estimate of mean percent of individuals subject to removal (harvest) annually
- **nPlots_TREE**: number of non-zero plots used to compute tree estimates
- **nPlots_AREA**: number of non-zero plots used to compute land area estimates

Note

All sampling error estimates are returned as percentages, and represent ~68% confidence (1 standard deviation). To compute sampling error percent with 95% confidence, multiply by 1.96.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

Bechtold, W.A.; Patterson, P.L., eds. 2005. The Enhanced Forest Inventory and Analysis Program - National Sampling Design and Estimation Procedures. Gen. Tech. Rep. SRS - 80. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 85 p. https://www.srs.fs.usda.gov/pubs/gtr/gtr_srs080/gtr_srs080.pdf

See Also

tpa, vitalRates

Examples

```
## Not run:

## Load data from rFIA package
library(rFIA)
data(fiaRI)
data(countiesRI)


## Most recent estimates for all stems on forest land by species
growMort(db = clipFIA(fiaRI, mostRecent = TRUE),
         landType = 'forest',
         bySpecies = TRUE)

## Same as above, but implemented in parallel (much quicker)
detectCores() # 4 cores available, we will take 3
growMort(db = clipFIA(fiaRI, mostRecent = TRUE),
         landType = 'forest',
         bySpecies = TRUE,
         nCores = 3)

## Most recent estimates for growing-stock on timber land by species
growMort(db = clipFIA(fiaRI, mostRecent = TRUE),
         landType = 'timber',
         treeType = 'gs',
         bySpecies = TRUE)

## Estimates for all stems greater than 20 in DBH on forestland for all
##  available inventories (time-series)
growMort(db = fiaRI,
         landType = 'forest',
         treeDomain = DIA > 20)

## Most recent estimates grouped by stand age on forest land
# Make a categorical variable which represents stand age (grouped by 10 yr intervals)
fiaRI$COND$STAND_AGE <- makeClasses(fiaRI$COND$STDAGE, interval = 10, lower = 0)
growMort(db = clipFIA(fiaRI, mostRecent = TRUE),
         grpBy = 'STAND_AGE')

## Estimates for all white pine ( > 12" DBH) on forested mesic sites (all available inventories)
growMort(fiaRI,
         treeDomain = SPCD == 129 & DIA > 12, # Species code for white pine
         areaDomain = PHYSCLCD %in% 21:29) # Mesic Physiographic classes

## Most recent estimates for all stems on forest land grouped by user-defined areal units
ctSF <- growMort(clipFIA(fiaRI, mostRecent = TRUE),
                 polys = countiesRI,
                 returnSpatial = TRUE)
```

```
plot(ctSF) # Plot multiple variables simultaneously
plotFIA(ctSF, 'MORT_TPA') # Plot of Mortality TPA with color scale

## End(Not run)
```

`invasive` *Estimate invasive species coverage from FIADB*

Description

Produces estimates of areal coverage of invasive species from the Forest Inventory and Analysis Database. Estimates can be produced for regions defined within the FIA Database (e.g. counties), at the plot level, or within user-defined areal units. All estimates are returned by species although can be grouped by other variables defined in the FIADB. If multiple reporting years (EVALIDs) are included in the data, estimates will be output as a time series. If multiple states are represented by the data, estimates will be output for the full region (all area combined), unless specified otherwise (e.g. `grpBy = 'STATECD'`). Easy options to implement parallel processing.

Usage

```
invasive(db, grpBy = NULL, polys = NULL, returnSpatial = FALSE, landType = "forest",
         areaDomain = NULL, byPlot = FALSE, totals = FALSE, SE = TRUE,
         progress = TRUE, nCores = 1)
```

Arguments

`db`	`FIA.Database` object produced from `readFIA`. Function requires that PLOT, INVASIVE_SUBPLOT_SPP, COND, POP_PLOT_STRATUM_ASSGN, POP_ESTN_UNIT, POP_EVAL, POP_STRATUM, POP_EVAL_TYP, POP_EVAL_GRP tables exist in FIA.Database object.
`grpBy`	character; variables from PLOT, COND, or TREE tables to group estimates by. Multiple grouping variables should be combined with `c()`, and grouping will occur heirarchically. For example, to produce seperate estimates for each ownership group within ecoregion subsections, specify `c('ECOSUBCD','OWNGRPCD')`.
`polys`	`sp` or `sf` Polygon/MultiPolgyon object; Areal units to bin data for estimation. Seperate estimates will be produces for region encompassed by each areal unit.
`returnSpatial`	logical; if TRUE, return sf spatial object (polys must also be specified).
`landType`	character ('forest' or 'timber'); Type of land which estimates will be produced for. Timberland is a subset of forestland (default) which has high site potential and non-reserve status (see details).
`areaDomain`	logical predicates defined in terms of the variables in PLOT and/or COND tables. Used to define the area for which estimates will be produced (e.g. within 1 mile of improved road: `RDDISTCD %in% c(1:6)`, Hard maple/basswood forest type: `FORTYPCD == 805`). Multiple conditions are combined with `&` (and) or `|` (or). Only plots within areas where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.

byPlot	logical; if TRUE, returns estimates for individual plot locations (population totals not returned).
totals	logical; if TRUE, returns population estimates in addition to % coverage.
SE	logical; if TRUE, returns estimates with samping error (approx. 5x faster without returning samping errors)
progress	logical; if TRUE, a dynamic progress bar is displayed (approx. 30% faster when FALSE.
nCores	numeric; number of cores to use for parallel implementation. Check available cores using detectCores. Default = 1, serial processing.

Details

Estimation of attributes follows the procedures documented in Bechtold and Patterson (2005). Specifically, percent areal coverage is computed using a sample-based ratio-of-means estimator of total invasive coverage area / total land area within the domain of interest. Estimates of areal coverage of individual invasive species should NOT be summed to produce estimates of areal coverage by ALL invasive species, as areal coverage by species is not mutually exclusive (multiple species my occur in the same area). Current FIA data collection protocols do not allow for the unbiased estimation of areal coverage by all invasive species.

Stratified random sampling techniques are most often employed to compute estimates in recent inventories, although double sampling and simple random sampling may be employed for early inventories. Estimates are adjusted for non-response bias by assuming attributes of non-response plot locations to be equal to the mean of other plots included within thier respective stratum or population.

Forest land must be at least 10-percent stocked by trees of any size, including land that formerly had such tree cover and that will be naturally or artificially regenerated. Forest land includes transition zones, such as areas between heavily forested and nonforested lands that are at least 10-percent stocked with trees and forest areas adjacent to urban and builtup lands. The minimum area for classification of forest land is 1 acre and 120 feet wide measured stem-to-stem from the outermost edge. Unimproved roads and trails, streams, and clearings in forest areas are classified as forest if less than 120 feet wide. Timber land is a subset of forest land that is producing or is capable of producing crops of industrial wood and not withdrawn from timber utilization by statute or administrative regulation. (Note: Areas qualifying as timberland are capable of producing at least 20 cubic feet per acre per year of industrial wood in natural stands. Currently inaccessible and inoperable areas are NOT included).

Easy, efficient parallelization is implemented with the parallel package. Users must only specify the nCores argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (nCores = 1).

Value

Dataframe or SF object (if returnSpatial = TRUE). If byPlot = TRUE, values of areal coverage are returned for each plot. All variables with names ending in SE, represent the estimate of sampling error (%) of the variable.

- **YEAR**: reporting year associated with estimates
- **SYMBOL**: unique species ID from NRCS Plant Reference Guide
- **SCIENTIFIC_NAME**: scientific name of the species
- **COMMON_NAME**: common name of the species
- **COVER_PCT**: estimate of percent areal coverage of the species
- **COVER_AREA**: estimate of areal coverage of the species (acres)
- **AREA**: estimate of total land area (acres)
- **nPlots_INV**: number of non-zero plots used to compute invasive coverage estimates
- **nPlots_AREA**: number of non-zero plots used to compute land area estimates

Note

All sampling error estimates are returned as percentages, and represent ~68% confidence (1 standard deviation). To compute sampling error percent with 95% confidence, multiply by 1.96.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

Bechtold, W.A.; Patterson, P.L., eds. 2005. The Enhanced Forest Inventory and Analysis Program - National Sampling Design and Estimation Procedures. Gen. Tech. Rep. SRS - 80. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 85 p. https://www.srs.fs.usda.gov/pubs/gtr/gtr_srs080/gtr_srs080.pdf

See Also

dwm, tpa, clipFIA

Examples

```
## Not run:

## Load data from rFIA package
library(rFIA)
data(fiaRI)
data(countiesRI)


## Most recent estimates on forest land
invasive(db = clipFIA(fiaRI),
    landType = 'forest')

## Same as above, but implemented in parallel (much quicker)
detectCores() # 4 cores available, we will take 3
invasive(db = clipFIA(fiaRI),
```

```
         landType = 'forest',
         nCores = 3)

## Same as above, but return population totals as well (total areal coverage, total forest area)
invasive(db = clipFIA(fiaRI, mostRecent = TRUE),
    landType = 'forest',
    totals = TRUE)

## Estimates for all available inventories (time-series) on forestland,
##  with no sampling error estimates (~5x faster)
invasive(db = fiaRI, SE = FALSE)

## Most recent estimates grouped by stand age on forest land
# Make a categorical variable which represents stand age (grouped by 10 yr intervals)
fiaRI$COND$STAND_AGE <- makeClasses(fiaRI$COND$STDAGE, interval = 10)
invasive(db = clipFIA(fiaRI, mostRecent = TRUE),
         grpBy = 'STAND_AGE')

## Estimates on forested mesic sites (all available inventories)
invasive(fiaRI,
         areaDomain = PHYSCLCD %in% 21:29) # Mesic Physiographic classes

## Most recent estimates on forest land grouped by user-defined areal units
ctSF <- invasive(clipFIA(fiaRI, mostRecent = TRUE),
                 polys = countiesRI,
                 returnSpatial = TRUE)
plot(ctSF) # Plot multiple variables simultaneously
plotFIA(ctSF[ctSF$SYMBOL == 'ROMU',], 'COVER_PCT') # Plot Multiflora rose coverage

## End(Not run)
```

`makeClasses` *Convert numeric variables to class intervals (factor)*

Description

Convert continuous numeric variables to class intervals with output as factor or numeric classes. Simplified implementation of `cut`. Example uses include computing diameter or height classes for summarization with **rFIA** functions (e.g. `tpa`, `biomass`).

Usage

```
makeClasses(x, interval = NULL, lower = NULL, upper = NULL,
            brks = NULL, numLabs = FALSE)
```

Arguments

`x`	numeric vector to be converted to factor (class intervals).
`interval`	numeric; interval of desired output classes. e.g. specify `x = DIA` and `interval = 2` for 2-inch diameter class intervals.

lower	lower bound of output classes, included in lowest class. e.g. [lower, ...).
upper	upper bound of output classes, NOT included in highest class. e.g. [..., upper).
brks	numeric vector of desired breakpoints (bounds) of class intervals.
numLabs	logical; if TRUE, return class intervals as numeric vector with values representing the lower bounds of each interval. If FALSE, return factor with labels of form '[b1,b2)'.

Value

Factor or integer vector. Factor values represent class intervals with [b1,b2) notation, values of integer vectors represent the lower bounds of class intervals (e.g. b1).

Author(s)

Hunter Stanke and Andrew Finley

See Also

clipFIA

Examples

```
## Load data from the rFIA package
library(rFIA)
data(fiaRI)

## Compute Diameter Classes on 1-inch intervals for each tree in TREE table ----
# Factor w/ interval labels
makeClasses(fiaRI$TREE$DIA, interval = 1)
# Numeric w/ lower bound of each class as returned value
makeClasses(fiaRI$TREE$DIA, interval = 1, numLabs = TRUE)

## Compute Stand Age Classes on 20 year intervals for each
## condition in COND table ----
# NOTE: Unrecorded stand age recorded as -999, replace negative values with NA
fiaRI$COND$STDAGE[fiaRI$COND$STDAGE < 0] <- NA
makeClasses(fiaRI$COND$STDAGE, interval = 25)

## Compute Stand Stocking Classes (10%) for all live (ALSTK),
## and growing stock (GSSTK) in COND table ----
makeClasses(fiaRI$COND$ALSTK, interval = 10) # All Live
makeClasses(fiaRI$COND$GSSTK, interval = 10) # Growing Stock

## Compute % Slope Classes (20%) for each condition in COND table ----
makeClasses(fiaRI$COND$SLOPE, interval = 20)
```

plotFIA *Static and animated plots of spatial FIA summaries*

Description

Produce static and temporally animated plots of spatial FIA summaries. Requires that summaries be returned as spatial objects (e.g. specify `returnSpatial = TRUE` when computing summaries using `tpa`). For animated plots, also requires that multiple reporting years be present in the summary data (animations iterate through time). Specify `savePath` and `fileName` to save plots (animations saved as .gif files).

Usage

```
plotFIA(data, fillVar, animate = FALSE, title = NULL, colOption = 'viridis',
        lineCol = "gray30", lineWidth =1, minYear = 2005, direction = 1,
        alpha = .9, transform = "identity", text.size = 1, text.font = '',
        lab.width = 1, legend.height = 1, legend.width = 1, device = NULL,
        savePath = NULL, fileName = NULL, ...)
```

Arguments

`data`	sf object; spatial FIA summary produced from other `rFIA` functions (e.g. `tpa`, `biomass`, etc.).
`fillVar`	character; variable contained in `data` which will be used to fill polygons. Variable name must be quoted.
`animate`	logical; if TRUE, produces temporally animated plots (iterates over years).
`title`	character; plot title.
`colOption`	character; one of: "viridis" (default), "magma", "inferno", "plasma", or "cividis".
`lineCol`	character; polygon outline color.
`lineWidth`	numeric; scalar for polygon outline width. Specify `lineWidth = 0` for no outline.
`minYear`	numeric; earliest year to be included in animation. FIA data is sparse in years prior to 2005 and estimates are unlikely to be available.
`direction`	numeric; sets the order of colors in the scale. If 1, the default, colors are ordered from darkest to lightest. If -1, the order of colors is reversed.
`alpha`	numeric; alpha transparency, a number in [0,1], see argument alpha in `hsv`.
`transform`	character; transformations to apply to plotted variable `fillVar`. Options include: "asn", "atanh", "boxcox", "exp", "identity", "log", "log10", "log1p", "log2", "logit", "reciprocal", "reverse", "sqrt".
`text.size`	numeric; scalar for text size (e.g. text.size = 2 would be twice the default size).

text.font	character; font family. Choose from: 'Short', 'Canonical', 'mono', 'Courier', 'sans', 'Helvetica', 'serif', 'Times', 'AvantGarde', 'Bookman', 'Helvetica-Narrow', 'NewCenturySchoolbook', 'Palatino', 'URWGothic', 'URWBookman', 'NimbusMon', 'URWHelvetica', 'NimbusSan', 'NimbusSanCond', 'CenturySch', 'URWPalladio', 'URWTimes', or 'NimbusRom'.
lab.width	numeric; scalar for legend title width. This value controls text wrapping in title.
legend.height	numeric; scalar for legend height.
legend.width	numeric; scalar for legend width.
device	character; device to use for image save. Can either be a device function (e.g. png()), or one of "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "png", "bmp", "svg" or "wmf" (windows only)..
savePath	character; path to save plot to (combined with fileName).
fileName	character; file name to create on disk.
...	other arguments passed on to scale_fill_viridis_c.

Author(s)

Hunter Stanke and Andrew Finley

Examples

```
## Not run:
## Compute abundance estimates for live stems in Rhode Island
## for all available inventory years, summarized by counties
tpaRI <- tpa(fiaRI, polys = countiesRI, returnSpatial = TRUE)

## Produce animated plot
plotFIA(tpaRI, fillVar = 'TPA', animate = TRUE, title = 'Abundance (TPA)')

## With a square root transform
plotFIA(tpaRI, fillVar = 'TPA', animate = TRUE, title = 'Abundance (TPA)', transform = 'sqrt')


## Same as above, but for static plots (most recent subset from RI)
tpaMR <- tpa(clipFIA(fiaRI), polys = countiesRI, returnSpatial = TRUE)

## Produce animated plot
plotFIA(tpaMR, fillVar = 'TPA', animate = FALSE, title = 'Abundance (TPA)')

## With a square root transform
plotFIA(tpaMR, fillVar = 'TPA', animate = FALSE, title = 'Abundance (TPA)', transform = 'sqrt')

## End(Not run)
```

readFIA *Load FIA database into R environment*

Description

Loads FIA Datatables into R from .csv files stored in a local directory. Capable of merging multiple state downloads of the FIA database for regional analysis. Simply store each set of state data, as download from the FIA Datamart, in the same directory and hand to readFIA. Easy options to implement parallel processing.

Usage

```
readFIA(dir, common = TRUE, tables = NULL, nCores = 1, ...)
```

Arguments

dir	directory where FIA Datatables are stored.
common	logical; if TRUE, only import most commonly used tables, including all required for rFIA functions (see Details for list of tables).
tables	character vector; names of specific tables to be imported (e.g. 'PLOT', 'TREE', 'COND', 'TREE_GRM_COMPONENT').
nCores	numeric; number of cores to use for parallel implementation. Check available cores using detectCores. Default = 1, serial processing.
...	other arguments to pass to read.csv.

Details

Download subsets of the FIA Database from the FIA Datamart: https://apps.fs.usda.gov/fia/datamart/datamart.html. Once downloaded, unzip the directory, and read into R using readFIA.

If common = TRUE, the following tables will be imported: COND, COND_DWM_CALC, INVASIVE_SUBPLOT_SPP, PLOT, POP_ESTN_UNIT, POP_EVAL, POP_EVAL_GRP, POP_EVAL_TYP, POP_PLOT_STRATUM_ASSGN, POP_STRATUM, SUBPLOT, TREE, TREE_GRM_COMPONENT. These tables currently support all functionality with rFIA, and it is recommended that only these tables be imported to conserve RAM and reduce processing time.

If you wish to merge multiple state downloads of FIA data (e.g. Michigan and Indiana state downloads), simply place both sets of datatables in the same directory and import with readFIA. Upon import, corresponding tables (e.g. MI_PLOT and IN_PLOT) will be merged, and analysis can be completed for the entire region or within spatial units which transcend state boundaries (e.g. Ecoregion subsections).

Value

List object containing FIA Datatables. List elements represent individual FIA Datatables stored as `data.frame` objects. Names of list elements reflect names of files from which they were read into R environment (File names should not be changed after download from FIA Datamart).

If multiple subsets of the FIA database are held in the same directory (e.g. Michigan and Indiana state downloads), corresponding tables will be merged (e.g. PLOT table returned contains plots in both Michigan and Indiana).

Easy, efficient parallelization is implemented with the `parallel` package. Users must only specify the `nCores` argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (`nCores = 1`).

Note

To download subsets of the FIA Database, go online to the FIA Datamart (https://apps.fs.usda.gov/fia/datamart/datamart.html) and choose to download .csv files. Here you can choose to download subsets of the full database for individual states, or select to download individual tables. For use with the `rFIA` package, we recommend download of subsets of the full database representing individual states of interest. Files must be unzipped in order to be imported.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA DataMart: https://apps.fs.usda.gov/fia/datamart/datamart.html

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

See Also

`clipFIA`

Examples

```
## Not run:
library(rFIA)

## Read in database from local directory (files are as download from FIA Datamart)
db <- readFIA(dir = 'path/to/files', common = TRUE)

## Access individual tables from the FIA Database
# Plot table
db$PLOT
```

```
# Tree table
db$TREE

# Condition table
db$COND


## End(Not run)
```

`standStruct` *Estimate forest structural stage distribution from FIADB*

Description

Estimates the stand structural stage distribution of an area of forest/ timberland from FIA data. Estimates can be produced for regions defined within the FIA Database (e.g. counties), at the plot level, or within user-defined areal units. If multiple reporting years (EVALIDs) are included in the data, estimates will be output as a time series. Easy options to implement parallel processing. Stand structural stage is classified for each stand (condition) using a method similar to that of Frelich and Lorimer (1991) but substitute basal area for exposed crown area (see Details, References). If multiple states are represented by the data, estimates will be output for the full region (all area combined), unless specified otherwise (e.g. `grpBy = 'STATECD'`).

Usage

```
standStruct(db, grpBy = NULL, polys = NULL, returnSpatial = FALSE,
            landType = 'forest', areaDomain = NULL,
            byPlot = FALSE, totals = FALSE, tidy = TRUE, SE = TRUE,
            progress = TRUE, nCores = 1)
```

Arguments

`db`	`FIA.Database` object produced from `readFIA`. Function requires that PLOT, TREE, COND, POP_PLOT_STRATUM_ASSGN, POP_ESTN_UNIT, POP_EVAL, POP_STRATUM, POP_EVAL_TYP, POP_EVAL_GRP tables exist in FIA.Database object.
`grpBy`	character; variables from PLOT, COND, or TREE tables to group estimates by. Multiple grouping variables should be combined with `c()`, and grouping will occur heirarchically. For example, to produce seperate estimates for each ownership group within ecoregion subsections, specify `c('ECOSUBCD','OWNGRPCD')`.
`polys`	`sp` or `sf` Polygon/MultiPolgyon object; Areal units to bin data for estimation. Seperate estimates will be produces for region encompassed by each areal unit.
`returnSpatial`	logical; if TRUE, return sf spatial object (polys must also be specified).

landType	character ('forest' or 'timber'); Type of land which estimates will be produced for. Timberland is a subset of forestland (default) which has high site potential and non-reserve status (see details).
areaDomain	logical predicates defined in terms of the variables in PLOT and/or COND tables. Used to define the area for which estimates will be produced (e.g. within 1 mile of improved road: `RDDISTCD %in% c(1:6)`, Hard maple/basswood forest type: `FORTYPCD == 805`). Multiple conditions are combined with `&` (and) or `|` (or). Only plots within areas where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
byPlot	logical; if TRUE, returns estimates for individual plot locations (population totals not returned).
totals	logical; if TRUE, returns population estimates in addition to ratios.
tidy	logical; if TRUE, returns estimates grouped by structural stage, rather than including individual columns for each stand structural stage (For use in tidyverse packages, e.g. `ggplot2`, `dplyr`). Not recommended when returning spatial objects (`returnSpatial = TRUE`), for consistency with shapefile data structures.
SE	logical; if TRUE, returns estimates with samping error (approx. 5x faster without returning samping errors)
progress	logical; if TRUE, a dynamic progress bar is displayed (approx. 30% faster when FALSE.
nCores	numeric; number of cores to use for parallel implementation. Check available cores using `detectCores`. Default = 1, serial processing.

Details

Estimation of attributes follows the procedures documented in Bechtold and Patterson (2005). Specifically, the percent land area occupied by forest in each stand structural stage are computed using a sample-based ratio-of-means estimator of total area in structural stage / total land area within the domain of interest. Stand structural stage is classified based on the relative basal area of canopy stems in various size classes (defined below). Only stems which are identified on-site dominant, subdominant, or intermdediate crown-classes are used to classify stand structural stage.

Diameter Classes

- *Pole*: 11 - 25.9 cm
- *Mature*: 26 - 45.9 cm
- *Large*: 46+ cm

Structural Stage Classification

- *Pole Stage*: > 67% BA in pole and mature classes, with more BA in pole than mature.
- *Mature Stage*: > 67% BA in pole and mature classes, with more BA in mature than pole OR > 67% BA in mature and large classes, with more BA in mature.
- *Late-Successional Stage*: > 67% BA in mature and large classes, with more in large
- *Mosiac*: Any plot not meeting above criteria.

Stratified random sampling techniques are most often employed to compute estimates in recent inventories, although double sampling and simple random sampling may be employed for early inventories. Estimates are adjusted for non-response bias by assuming attributes of non-response plot locations to be equal to the mean of other plots included within thier respective stratum or population.

Forest land must be at least 10-percent stocked by trees of any size, including land that formerly had such tree cover and that will be naturally or artificially regenerated. Forest land includes transition zones, such as areas between heavily forested and nonforested lands that are at least 10-percent stocked with trees and forest areas adjacent to urban and builtup lands. The minimum area for classification of forest land is 1 acre and 120 feet wide measured stem-to-stem from the outermost edge. Unimproved roads and trails, streams, and clearings in forest areas are classified as forest if less than 120 feet wide. Timber land is a subset of forest land that is producing or is capable of producing crops of industrial wood and not withdrawn from timber utilization by statute or administrative regulation. (Note: Areas qualifying as timberland are capable of producing at least 20 cubic feet per acre per year of industrial wood in natural stands. Currently inaccessible and inoperable areas are NOT included).

Easy, efficient parallelization is implemented with the `parallel` package. Users must only specify the `nCores` argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (`nCores = 1`).

Value

Dataframe or SF object (if `returnSpatial = TRUE`). If `byPlot = TRUE`, values of areal coverage are returned for each plot. All variables with names ending in `SE`, represent the estimate of sampling error (%) of the variable.

- **STAGE**: Stand structural stage.
- **PERC_AREA**: % land area in each structural stage.
- **TOTAL_AREA**: Total land area used to produce estimates (forest or timberland) (acres).

Note

All sampling error estimates are returned as percentages, and represent ~68% confidence (1 standard deviation). To compute sampling error percent with 95% confidence, multiply by 1.96.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

Bechtold, W.A.; Patterson, P.L., eds. 2005. The Enhanced Forest Inventory and Analysis Program - National Sampling Design and Estimation Procedures. Gen. Tech. Rep. SRS - 80. Asheville,

NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 85 p. https://www.srs.fs.usda.gov/pubs/gtr/gtr_srs080/gtr_srs080.pdf

Frelich, L. E., and Lorimer, C. G. (1991). Natural Disturbance Regimes in Hemlock-Hardwood Forests of the Upper Great Lakes Region. Ecological Monographs, 61(2), 145-164. doi:10.2307/1943005

Goodell, L., and Faber-Langendoen, D. (2007). Development of stand structural stage indices to characterize forest condition in Upstate New York. Forest Ecology and Management, 249(3), 158-170. doi:10.1016/j.foreco.2007.04.052

See Also

tpa, diversity

Examples

```
## Not run:

## Load data from rFIA package
library(rFIA)
data(fiaRI)
data(countiesRI)

## Calculate structural stage distribution of all forestland, over time
standStruct(fiaRI)

## Same as above, but implemented in parallel
detectCores() # 4 cores available, we will take 3
standStruct(fiaRI, nCores = 3)

## Same as above, but output contains seperate column for each structural stage,
##   rathern than grouping variable
standStruct(fiaRI, tidy = FALSE)

## Calculate structural stage distribution of all forestland by owner group, over time
standStruct(clipFIA(fiaRI, mostRecent = TRUE),
            grpBy = c('OWNGRPCD'))

## Calculate structural stage distribution of all forestland on xeric sites, over time
standStruct(clipFIA(fiaRI, mostRecent = TRUE),
            areaDomain = PHYSCLCD %in% c(11:19))

## Calculate structural stage distribution of all forestland by county and return
#   return spatial object
sdSF <- standStruct(clipFIA(fiaRI, mostRecent = TRUE),
            polys = countiesRI,
            returnSpatial = TRUE,
            tidy = FALSE)
plot(sdSF)
plotFIA(sdSF, 'POLE_PERC')

## End(Not run)
```

tpa *Estimate trees per acre and basal area per acre from FIADB*

Description

Produces tree per acre (TPA) and basal area per acre (BAA) estimates from FIA data, along with population totals for each variable. Estimates can be produced for regions defined within the FIA Database (e.g. counties), at the plot level, or within user-defined areal units. Options to group estimates by species, size class, and other variables defined in the FIADB. If multiple reporting years (EVALIDs) are included in the data, estimates will be output as a time series. If multiple states are represented by the data, estimates will be output for the full region (all area combined), unless specified otherwise (e.g. `grpBy = 'STATECD'`). Easy options to implement parallel processing.

Usage

```
tpa(db, grpBy = NULL, polys = NULL, returnSpatial = FALSE, bySpecies = FALSE,
    bySizeClass = FALSE, landType = 'forest', treeType = 'live',
    treeDomain = NULL, areaDomain = NULL, totals = FALSE,
    byPlot = FALSE, SE = TRUE, progress = TRUE, nCores = 1)
```

Arguments

db	`FIA.Database` object produced from `readFIA`. Function requires that PLOT, TREE, COND, POP_PLOT_STRATUM_ASSGN, POP_ESTN_UNIT, POP_EVAL, POP_STRATUM, POP_EVAL_TYP, POP_EVAL_GRP tables exist in FIA.Database object.
grpBy	character; variables from PLOT, COND, or TREE tables to group estimates by. Multiple grouping variables should be combined with `c()`, and grouping will occur heirarchically. For example, to produce seperate estimates for each ownership group within ecoregion subsections, specify `c('ECOSUBCD','OWNGRPCD')`.
polys	`sp` or `sf` Polygon/MultiPolgyon object; Areal units to bin data for estimation. Seperate estimates will be produces for region encompassed by each areal unit.
returnSpatial	logical; if TRUE, return sf spatial object (polys must also be specified).
bySpecies	logical; if TRUE, returns estimates grouped by species.
bySizeClass	logical; if TRUE, returns estimates grouped by size class (2-inch intervals, see `makeClasses` to compute different size class intervals).
landType	character ('forest' or 'timber'); Type of land which estimates will be produced for. Timberland is a subset of forestland (default) which has high site potential and non-reserve status (see details).
treeType	character ('all', 'live', 'dead', or 'gs'); Type of tree which estimates will be produced for. All (default) includes all stems, live and dead, greater than 1 in. DBH. Live/Dead includes all stems greater than 1 in. DBH which are live or dead (leaning less than 45 degrees), respectively. GS (growing-stock) includes live stems greater than 5 in. DBH which contain at least one 8 ft merchantable log.

treeDomain	logical predicates defined in terms of the variables in PLOT, TREE, and/or COND tables. Used to define the type of trees for which estimates will be produced (e.g. DBH greater than 20 inches: `DIA > 20`, Dominant/Co-dominant crowns only: `CCLCD %in% c(2,3)`). Multiple conditions are combined with `&` (and) or `|` (or). Only trees where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
areaDomain	logical predicates defined in terms of the variables in PLOT and/or COND tables. Used to define the area for which estimates will be produced (e.g. within 1 mile of improved road: `RDDISTCD %in% c(1:6)`, Hard maple/basswood forest type: `FORTYPCD == 805`). Multiple conditions are combined with `&` (and) or `|` (or). Only plots within areas where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
totals	logical; if TRUE, return population estimates (e.g. total area, total trees) along with ratio estimates (e.g. mean trees per acre).
byPlot	logical; if TRUE, returns estimates for individual plot locations (population totals not returned).
SE	logical; if TRUE, returns estimates with samping error (approx. 5x faster without returning samping errors)
progress	logical; if TRUE, a dynamic progress bar is displayed (approx. 30% faster when FALSE.
nCores	numeric; number of cores to use for parallel implementation. Check available cores using `detectCores`. Default = 1, serial processing.

Details

Estimation of attributes follows the procedures documented in Bechtold and Patterson (2005). Specifically, TPA and BAA are computed using a sample-based ratio-of-means estimator of total trees (BA) / total land area within the domain of interest. Percentages of TPA and BAA in the domain of interest are represented as the total number of trees of a particular type (live, white pine) / total number of trees (live and dead, all species) within the region. The total populations used to compute these percentages will not change by changing treeType, but will vary if the user specifies an areaDomain or treeDomain.

Stratified random sampling techniques are most often employed to compute estimates in recent inventories, although double sampling and simple random sampling may be employed for early inventories. Estimates are adjusted for non-response bias by assuming attributes of non-response plot locations to be equal to the mean of other plots included within thier respective stratum or population.

Forest land must be at least 10-percent stocked by trees of any size, including land that formerly had such tree cover and that will be naturally or artificially regenerated. Forest land includes transition zones, such as areas between heavily forested and nonforested lands that are at least 10-percent stocked with trees and forest areas adjacent to urban and builtup lands. The minimum area for classification of forest land is 1 acre and 120 feet wide measured stem-to-stem from the outermost edge. Unimproved roads and trails, streams, and clearings in forest areas are classified as forest if less than 120 feet wide. Timber land is a subset of forest land that is producing or is capable of producing crops of industrial wood and not withdrawn from timber utilization by statute or administrative regulation. (Note: Areas qualifying as timberland are capable of producing at

least 20 cubic feet per acre per year of industrial wood in natural stands. Currently inaccessible and inoperable areas are NOT included).

Easy, efficient parallelization is implemented with the `parallel` package. Users must only specify the `nCores` argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (`nCores = 1`).

Value

Dataframe or SF object (if `returnSpatial = TRUE`). If `byPlot = TRUE`, values are returned for each plot. All variables with names ending in `SE`, represent the estimate of sampling error (%) of the variable. All variables with names ending in `TOTAL` represent the population total of the variable.

- **YEAR**: reporting year associated with estimates
- **TPA**: estimate of mean trees per acre
- **BAA**: estimate of mean basal area (sq. ft.) per acre
- **TPA_PERC**: estimate of mean proportion of trees falling within the domain of interest, with respect to trees per acre
- **BAA_PERC**: estimate of mean proportion of trees falling within the domain of interest, with respect to basal area per acre
- **nPlots_TREE**: number of non-zero plots used to compute tree and basal area estimates
- **nPlots_AREA**: number of non-zero plots used to compute land area estimates

Note

All sampling error estimates are returned as percentages, and represent ~68% confidence (1 standard deviation). To compute sampling error percent with 95% confidence, multiply by 1.96.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

Bechtold, W.A.; Patterson, P.L., eds. 2005. The Enhanced Forest Inventory and Analysis Program - National Sampling Design and Estimation Procedures. Gen. Tech. Rep. SRS - 80. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 85 p. https://www.srs.fs.usda.gov/pubs/gtr/gtr_srs080/gtr_srs080.pdf

See Also

`biomass`, `growMort`

Examples

```
## Not run:

## Load data from rFIA package
library(rFIA)
data(fiaRI)
data(countiesRI)


## Most recent estimates for live stems on forest land by species
tpa(db = clipFIA(fiaRI, mostRecent = TRUE),
    landType = 'forest',
    treeType = 'live',
    bySpecies = TRUE)

## Most recent estimates for growing-stock on timber land by species
tpa(db = clipFIA(fiaRI, mostRecent = TRUE),
    landType = 'timber',
    treeType = 'gs',
    bySpecies = TRUE)

## Estimates for snags greater than 20 in DBH on forestland for all
##  available inventories (time-series)
tpa(db = fiaRI,
    landType = 'forest',
    treeType = 'dead',
    treeDomain = DIA > 20)

## Same as above, but implemented in parallel (much quicker)
detectCores() # 4 cores available, we will take 3
tpa(db = fiaRI,
    landType = 'forest',
    treeType = 'dead',
    treeDomain = DIA > 20,
    nCores = 3)

## Most recent estimates grouped by stand age on forest land
# Make a categorical variable which represents stand age (grouped by 10 yr intervals)
fiaRI$COND$STAND_AGE <- makeClasses(fiaRI$COND$STDAGE, interval = 10)
tpa(db = clipFIA(fiaRI, mostRecent = TRUE),
    grpBy = 'STAND_AGE')

## Estimates for live white pine ( > 12" DBH) on forested mesic sites (all available inventories)
tpa(fiaRI,
    treeType = 'live',
    treeDomain = SPCD == 129 & DIA > 12, # Species code for white pine
    areaDomain = PHYSCLCD %in% 21:29) # Mesic Physiographic classes

## Most recent estimates for all stems on forest land grouped by user-defined areal units
ctSF <- tpa(clipFIA(fiaRI, mostRecent = TRUE),
            polys = countiesRI,
            returnSpatial = TRUE)
```

```
plot(ctSF) # Plot multiple variables simultaneously
plotFIA(ctSF, 'TPA') # Plot of TPA with color scale

## End(Not run)
```

`vitalRates` *Estimate tree growth rates from FIADB*

Description

Computes estimates of average annual DBH, basal area, height, and net volume growth rates for individual stems, along with average annual basal area and net volume growth per acre. Estimates can be produced for regions defined within the FIA Database (e.g. counties), at the plot level, or within user-defined areal units. Options to group estimates by species, size class, and other variables defined in the FIADB. If multiple reporting years (EVALIDs) are included in the data, estimates will be output as a time series. If multiple states are represented by the data, estimates will be output for the full region (all area combined), unless specified otherwise (e.g. `grpBy = 'STATECD'`). Easy options to implement parallel processing.

Usage

```
vitalRates(db, grpBy = NULL, polys = NULL, returnSpatial = FALSE, bySpecies = FALSE,
           bySizeClass = FALSE, landType = 'forest', treeType = 'live',
          treeDomain = NULL, areaDomain = NULL, totals = FALSE, byPlot = FALSE,
           SE = TRUE, progress = TRUE, nCores = 1)
```

Arguments

`db`	`FIA.Database` object produced from `readFIA`. Function requires that PLOT, TREE, TREE_GRM_COMPONENT, TREE_GRM_ESTN, COND, POP_PLOT_STRATUM_ASSGN, POP_ESTN_UNIT, POP_EVAL, POP_STRATUM, POP_EVAL_TYP, POP_EVAL_GRP, and SUBP_COND_CHNG_MTRX tables exist in FIA.Database object.
`grpBy`	character; variables from PLOT, COND, or TREE tables to group estimates by. Multiple grouping variables should be combined with `c()`, and grouping will occur heirarchically. For example, to produce seperate estimates for each ownership group within ecoregion subsections, specify `c('ECOSUBCD','OWNGRPCD')`.
`polys`	`sp` or `sf` Polygon/MultiPolgyon object; Areal units to bin data for estimation. Seperate estimates will be produces for region encompassed by each areal unit.
`returnSpatial`	logical; if TRUE, return sf spatial object (polys must also be specified).
`bySpecies`	logical; if TRUE, returns estimates grouped by species.
`bySizeClass`	logical; if TRUE, returns estimates grouped by size class (2-inch intervals, see `makeClasses` to compute different size class intervals).
`landType`	character ('forest' or 'timber'); Type of land which estimates will be produced for. Timberland is a subset of forestland (default) which has high site potential and non-reserve status (see details).

treeType	character ('live' or 'gs'); Type of tree which estimates will be produced for. All (default) includes all stems, live and dead, greater than 1 in. DBH. Live/Dead includes all stems greater than 1 in. DBH which are live or dead (leaning less than 45 degrees), respectively. GS (growing-stock) includes live stems greater than 5 in. DBH which contain at least one 8 ft merchantable log.
treeDomain	logical predicates defined in terms of the variables in PLOT, TREE, and/or COND tables. Used to define the type of trees for which estimates will be produced (e.g. DBH greater than 20 inches: `DIA > 20`, Dominant/Co-dominant crowns only: `CCLCD %in% c(2,3))`. Multiple conditions are combined with `&` (and) or `|` (or). Only trees where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
areaDomain	logical predicates defined in terms of the variables in PLOT and/or COND tables. Used to define the area for which estimates will be produced (e.g. within 1 mile of improved road: `RDDISTCD %in% c(1:6)`, Hard maple/basswood forest type: `FORTYPCD == 805)`. Multiple conditions are combined with `&` (and) or `|` (or). Only plots within areas where the condition evaluates to TRUE are used in producing estimates. Should NOT be quoted.
totals	logical; if TRUE, return population estimates (e.g. total area, total BA growth) along with ratio estimates (e.g. mean BA growth per acre).
byPlot	logical; if TRUE, returns estimates for individual plot locations (population totals not returned).
SE	logical; if TRUE, returns estimates with samping error (approx. 5x faster without returning samping errors)
progress	logical; if TRUE, a dynamic progress bar is displayed (approx. 30% faster when FALSE.
nCores	numeric; number of cores to use for parallel implementation. Check available cores using `detectCores`. Default = 1, serial processing.

Details

Estimation of attributes follows the procedures documented in Bechtold and Patterson (2005). Average annual diameter, basal area, height, and net volume growth of a stem is computed using a sample-based ratio of means estimator of total diameter (basal area, height, net volume) growth / total trees, and average annual basal area and net volume growth per acre is computed as total basal area (net volume) growth / total area. All estimates are returned as average annual rates. Only conditions which were forest in time 1 and in time 2 are included in estimates (excluding converted stands). Only stems 5 inches DBH or greater are included in estimates.

As estimates are of net growth rates, they may attain a negative value. Negative growth estimates most likely indicate a substantial change in some attribute of the tree or area between time 1 and time 2, which caused the attribute to decrease. For example, if a stem loses its top between time 1 and time 2, albiet remains alive during this interval, the stem will have a negative height growth rate (assuming regrowth in height does not exceed that lost after breakage).

This function does not employ the growth accounting method used by FIA EVALIDator at this time. Rather, we assume that values (e.g. basal area) do not shift across classified attributes (e.g. size class) between time 1 and time 2, and thus our results will differ slightly from those of EVALIDator.

Future development of this function aims to employ the growth accounting method used by FIA, and diversify the forms of our estimates (e.g. merchantable volume, biomass, carbon).

Stratified random sampling techniques are most often employed to compute estimates in recent inventories, although double sampling and simple random sampling may be employed for early inventories. Estimates are adjusted for non-response bias by assuming attributes of non-response plot locations to be equal to the mean of other plots included within thier respective stratum or population.

Forest land must be at least 10-percent stocked by trees of any size, including land that formerly had such tree cover and that will be naturally or artificially regenerated. Forest land includes transition zones, such as areas between heavily forested and nonforested lands that are at least 10-percent stocked with trees and forest areas adjacent to urban and builtup lands. The minimum area for classification of forest land is 1 acre and 120 feet wide measured stem-to-stem from the outermost edge. Unimproved roads and trails, streams, and clearings in forest areas are classified as forest if less than 120 feet wide. Timber land is a subset of forest land that is producing or is capable of producing crops of industrial wood and not withdrawn from timber utilization by statute or administrative regulation. (Note: Areas qualifying as timberland are capable of producing at least 20 cubic feet per acre per year of industrial wood in natural stands. Currently inaccessible and inoperable areas are NOT included).

Easy, efficient parallelization is implemented with the `parallel` package. Users must only specify the `nCores` argument with a value greater than 1 in order to implement parallel processing on their machines. Parallel implementation is achieved using a snow type cluster on any Windows OS, and with multicore forking on any Unix OS (Linux, Mac). Implementing parallel processing may substantially decrease free memory during processing, particularly on Windows OS. Thus, users should be cautious when running in parallel, and consider implementing serial processing for this task if computational resources are limited (`nCores = 1`).

Value

Dataframe or SF object (if `returnSpatial = TRUE`). If `byPlot = TRUE`, values are returned for each plot. All variables with names ending in `SE`, represent the estimate of sampling error (%) of the variable. All variables with names ending in `TOTAL` represent the population total of the variable.

- **YEAR**: reporting year associated with estimates
- **DIA_GROW**: estimate of mean annual diameter growth of a stem (inches/ yr)
- **BA_GROW**: estimate of mean annual basal area growth of a stem (sq. ft./ yr)
- **BAA_GROW**: estimate of mean annual basal area growth per acre (sq. ft./ acre/ yr)
- **HT_GROW**: estimate of mean annual basal area growth of a stem (ft./ yr)
- **NETVOL_GROW**: estimate of mean annual sound net volume growth of a stem (cu. ft./ yr)
- **NETVOL_GROW_AC**: estimate of mean annual sound net volume growth per acre (cu. ft./ acre/ yr)
- **nPlots_TREE**: number of non-zero plots used to compute tree and basal area estimates
- **nPlots_AREA**: number of non-zero plots used to compute land area estimates

Note

All sampling error estimates are returned as percentages, and represent ~68% confidence (1 standard deviation). To compute sampling error percent with 95% confidence, multiply by 1.96.

Author(s)

Hunter Stanke and Andrew Finley

References

FIA Database User Guide: https://www.fia.fs.fed.us/library/database-documentation/

Bechtold, W.A.; Patterson, P.L., eds. 2005. The Enhanced Forest Inventory and Analysis Program - National Sampling Design and Estimation Procedures. Gen. Tech. Rep. SRS - 80. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station. 85 p. https://www.srs.fs.usda.gov/pubs/gtr/gtr_srs080/gtr_srs080.pdf

See Also

growMort, tpa

Examples

```
## Not run:

## Load data from rFIA package
library(rFIA)
data(fiaRI)
data(countiesRI)


## Most recent estimates for live stems on forest land by species
vitalRates(db = clipFIA(fiaRI, mostRecent = TRUE),
           landType = 'forest',
           treeType = 'live',
           bySpecies = TRUE)

## Same as above, but implemented in parallel (much quicker)
detectCores() # 4 cores available, we will take 3
vitalRates(db = clipFIA(fiaRI, mostRecent = TRUE),
           landType = 'forest',
           treeType = 'live',
           bySpecies = TRUE,
           nCores = 3)

## Most recent estimates for growing-stock on timber land by species
vitalRates(db = clipFIA(fiaRI, mostRecent = TRUE),
           landType = 'timber',
           treeType = 'gs',
           bySpecies = TRUE)

## Estimates for stems greater than 20 in DBH on forestland for all
##  available inventories (time-series)
vitalRates(db = fiaRI,
           landType = 'forest',
           treeType = 'live',
           treeDomain = DIA > 20)
```

```
## Most recent estimates grouped by stand age on forest land
# Make a categorical variable which represents stand age (grouped by 10 yr intervals)
fiaRI$COND$STAND_AGE <- makeClasses(fiaRI$COND$STDAGE, interval = 10)
vitalRates(db = clipFIA(fiaRI, mostRecent = TRUE),
           grpBy = 'STAND_AGE')

## Estimates for live white pine ( > 12" DBH) on forested mesic sites
vitalRates(fiaRI,
           treeType = 'live',
           treeDomain = SPCD == 129 & DIA > 12, # Species code for white pine
           areaDomain = PHYSCLCD %in% 21:29) # Mesic Physiographic classes

## Most recent estimates for all stems on forest land grouped by user-defined areal units
ctSF <- vitalRates(clipFIA(fiaRI, mostRecent = TRUE),
                   polys = countiesRI,
                   returnSpatial = TRUE)
plot(ctSF) # Plot multiple variables simultaneously
plotFIA(ctSF, 'BAA_GROW') # Plot of BAA Growth with color scale

## End(Not run)
```

Index