

Package: forestTIME (via r-universe)

October 12, 2026

Title Generate Annualized Carbon and Biomass Estimates from FIA Data

Version 2.2.0.9000

Description Uses interpolation and extrapolation to create annualized versions of Forest Inventory Analysis (FIA).

License MIT + file LICENSE

URL <https://github.com/Evans-Ecology-Lab/forestTIME>
<https://cct-datascience.r-universe.dev/forestTIME>,
<https://evans-ecology-lab.github.io/forestTIME/>

BugReports <https://github.com/Evans-Ecology-Lab/forestTIME/issues>

Depends R (>= 4.2.0)

Imports cli (>= 3.6.4), curl (>= 6.2.1), dplyr (>= 1.1.4), fs (>= 1.6.5), glue (>= 1.8.0), Hmisc (>= 5.2-2), purrr (>= 1.0.4), rFIA (>= 1.1.1), rlang (>= 1.1.5), stringr (>= 1.5.1), tidyr (>= 1.3.1), vctrs

Suggests ggplot2, knitr, nanoparquet, quarto, readr, rmarkdown, testthat (>= 3.0.0), tibble, waldo, withr

VignetteBuilder knitr, quarto

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libuv1-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://cct-datascience.r-universe.dev>

Date/Publication 2026-10-12 18:50:10 UTC

RemoteUrl <https://github.com/Evans-Ecology-Lab/forestTIME>

RemoteRef HEAD

RemoteSha b69eeab004a61f3ee7b84cc0f60c5cd3bd2e5351

Contents

fia_add_composite_ids	2
fia_allometry	3
fia_annualize	4
fia_design	5
fia_download	6
fia_load	7
fia_split_composite_ids	7
fia_tidy	8
state_areas	9

Index	10
-------	----

fia_add_composite_ids	<i>Add composite ID columns to data</i>
-----------------------	---

Description

Creates a tree_ID and/or a plot_ID column that contain unique tree and plot identifiers, respectively. These are created by pasting together the values for UNITCD, STATECD, COUNTYCD, PLOT and in the case of trees SUBP and TREE.

Usage

```
fia_add_composite_ids(data)
```

Arguments

data	A tibble or data frame with at least the UNITCD, STATECD, COUNTYCD and PLOT columns
------	---

Value

The input tibble with a plot_ID and possibly also a tree_ID column added

See Also

See [fia_split_composite_ids\(\)](#) for "undoing" this.

Examples

```
db <- fia_load("RI", dir = system.file("exdata", package = "forestTIME"))
fia_add_composite_ids(db$TREE)
```

fia_allometry	<i>Scale interpolated tree-level biometric data to tree-level biomass and carbon using NSVB framework</i>
---------------	---

Description

Estimates tree-level biomass and carbon variables using the National Scale Volume and Biomass allometric equations (NSVB).

Usage

```
fia_allometry(data)
```

Arguments

`data` A data frame or tibble; generally the output of `fia_annualize()`.

Value

A tibble with the additional columns DRYBIO_AG and CARBON_AG that correspond to the FIAdB definitions of those variables.

Author(s)

David Walker

References

Westfall, J.A., Coulston, J.W., Gray, A.N., Shaw, J.D., Radtke, P.J., Walker, D.M., Weiskittel, A.R., MacFarlane, D.W., Affleck, D.L.R., Zhao, D., Temesgen, H., Poudel, K.P., Frank, J.M., Prisley, S.P., Wang, Y., Sánchez Meador, A.J., Auty, D., Domke, G.M., 2024. A national-scale tree volume, biomass, and carbon modeling system for the United States. U.S. Department of Agriculture, Forest Service. [doi:10.2737/wogtr104](https://doi.org/10.2737/wogtr104)

Examples

```
## Not run:
library(forestTIME)
db <- fia_load("RI")
data_tidy <- fia_tidy(db)
data_annualized <- fia_annualize(data_tidy)
data_carbon <- fia_allometry(data_annualized)

## End(Not run)
```

fia_annualize	Create annualized FIA data
---------------	----------------------------

Description

Converts tidied panel data into annualized data with interpolated measurements for trees for years between inventories. This happens in three steps, which can be "manually" replicated by chaining other forestTIME functions.

Usage

```
fia_annualize(data_tidy, year_var = c("INVYR", "MEASYEAR"), use_mortyr = TRUE)
```

Arguments

data_tidy	A tibble produced by fia_tidy() .
year_var	character indicating which year variable to use; default is INVYR.
use_mortyr	logical; Use MORTYR (if recorded) as the first year a tree was dead? Passed to adjust_mortality() .

Details

First, data is expanded by [expand_data\(\)](#) to add rows for years between inventories for each tree in the data. Next, data is interpolated with [interpolate_data\(\)](#). Finally, [adjust_mortality\(\)](#) is applied. For trees that die and/or fall between inventories, we adjust their history according either to a recorded MORTYR (if use_mortyr = TRUE) or, as a fall-back, the midpoint between surveys, rounded down. Unlike these intermediate functions, [fia_annualize\(\)](#) produces a dataset which can be safely used for other analyses (with the caveat that all of this is experimental).

Note

Most users should use this "wrapper" function rather than running each step separately since the intermediate steps may contain data artifacts. However, one reason to use the stepwise workflow would be to save time when generating interpolated data with and without using MORTYR as [interpolate_data\(\)](#) is the slowest step.

See Also

For more details on each step, see: [expand_data\(\)](#), [interpolate_data\(\)](#), [adjust_mortality\(\)](#)

Examples

```
## Not run:
db <- db <- fia_load("RI", dir = system.file("exdata", package = "forestTIME"))
data_tidy <- fia_tidy(db)
data_annualized <- fia_annualize(data_tidy)

## End(Not run)
```

fia_design

*Assign interpolated data plots to estimation units and strata***Description**

This adds columns from various POP_* tables to annualized data so that it can be used for stratified estimation (see, for example, [FIA Demystified](#)). Each plot in each year is matched with an EVALID and it's associated data as such:

1. The EVALID must have both the "EXPVOL" and "EXPCURR" EVAL_TYP.
2. The YEAR must be between START_INVYR and END_INVYR.
3. The first matching EVALID is used—i.e. if the YEAR is 2002 and there are EVALIDs for 2000–2003, 2001–2004, and 2002–2005, the one from 2000–2003 will be matched.

Usage

```
fia_design(data_annualized, db)
```

Arguments

data_annualized

Annualized data produced by [fia_annualize\(\)](#).

db

The list of tables produced by [fia_load\(\)](#).

Details

EVALID-associated data added by this function **may be in conflict with the results of interpolation by [fia_annualize\(\)](#)**! When using this function to do stratified estimation, use the PLOT_STATUS_CD as part of the domain indicator to correctly exclude any non-sampled plots with no tree data!

See Also

[fia_eval_info\(\)](#) to see all possible EVALIDs associated with plots.

Examples

```
## Not run:
# Load example data included in package
db <- fia_load("RI", dir = system.file("exdata", package = "forestTIME"))

# Annualize data
data_annualized <- db |> fia_tidy() |>
  fia_annualize(use_mortyr = FALSE)

# Assign plots to strata, estimation units, and EVALIDs
data_stratified <- fia_design(data_annualized, db)

## End(Not run)
```

fia_download

Download zip files from FIA datamart

Description

The zip files are smaller than just the *_TREE.csv, so this just downloads the whole zip and extracts the required CSV files. Uses `curl::multi_download()` which resumes partial skips already completed downloads when run subsequent times.

Usage

```
fia_download(
  states,
  download_dir = "fia",
  extract = c("forestTIME", "rFIA", "all", "none"),
  keep_zip = TRUE
)
```

Arguments

states	Vector of state abbreviations; for all states use <code>state.abb</code> .
download_dir	Where to save the zip files.
extract	Which files to extract from the downloaded zip file—those needed by <code>forestTIME</code> , those needed by <code>rFIA</code> (in addition to all the tables <code>forestTIME</code> needs), all the files, or none.
keep_zip	Logical; keep the .zip file after CSVs are extracted? Defaults to <code>TRUE</code> .

Value

Returns nothing.

Examples

```
## Not run:
# "Standard" download
fia_download(states = c("RI", "DE"))

# Extract enough tables so it works with rFIA::readFIA() also
fia_download(states = c("RI", "DE"), extract = "rFIA")

## End(Not run)
```

fia_load	<i>Read in needed tables</i>
----------	------------------------------

Description

Wrapper for `rFIA::readFIA()` that reads in the necessary tables

Usage

```
fia_load(states, dir = "fia")
```

Arguments

states	character; state/ US territory abbreviations (e.g. 'AL', 'MI', etc.) indicating which state subsets to read. Data for each state must be in dir. Choose to read multiple states by passing character vector of state abbreviations (e.g. states = c('RI', 'CT', 'MA')). If states = NULL, data for all states within dir will be read in and merged into a regional database.
dir	directory where .csv files of FIA tables are stored.

Value

A list of data frames

Examples

```
## Not run:
fia_download(states = "RI")
RI_db <- fia_load("RI")

## End(Not run)
```

fia_split_composite_ids	<i>Split composite ID columns</i>
-------------------------	-----------------------------------

Description

Splits the composite ID columns tree_ID and/or plot_ID into their original component columns

Usage

```
fia_split_composite_ids(data)
```

Arguments

`data` A tibble with the `tree_ID` and/or `plot_ID` columns

Value

The input tibble with additional columns `UNITCD`, `STATECD`, `COUNTYCD`, `PLOT` and possibly `SUBP` and `TREE`.

See Also

[fia_add_composite_ids\(\)](#)

Examples

```
db <- fia_load("RI", dir = system.file("exdata", package = "forestTIME"))
data_tidy <- fia_tidy(db)
fia_split_composite_ids(data_tidy)
```

`fia_tidy`

Read in and join all required tables

Description

Reads in all the tables needed for carbon estimation and population scaling and joins them into a single table. Then, some additional data cleaning steps are performed.

1. Creates unique tree and plot identifiers (`tree_ID` and `plot_ID`, respectively).
2. Fills in missing values for `ACTUALHT` with values from `HT` to prepare for interpolation.
3. Overwrites `SPCD` with whatever the last value of `SPCD` is for each tree (to handle trees that change `SPCD`).
4. Fills a tree's `MORTYR` column so every row contains the recorded mortality year.
5. Filters to only include inventories that came after the start of the annual inventory in each state.

Usage

```
fia_tidy(db)
```

Arguments

`db` A list of tables produced by [fia_load\(\)](#).

Value

A tibble.

See Also

[fia_add_composite_ids\(\)](#)

Examples

```
db <- fia_load("RI", dir = system.file("exdata", package = "forestTIME"))
data_tidy <- fia_tidy(db)
```

state_areas	<i>State codes used in FIA</i>
-------------	--------------------------------

Description

A data frame tha matches the STATECD column used in FIA data to the two letter state abbreviations and the total land area of the state in acres.

Usage

```
state_areas
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 50 rows and 3 columns.

Index

* datasets

state_areas, 9

adjust_mortality(), 4

expand_data(), 4

fia_add_composite_ids, 2

fia_add_composite_ids(), 8

fia_allometry, 3

fia_annualize, 4

fia_annualize(), 3, 5

fia_design, 5

fia_download, 6

fia_eval_info(), 5

fia_load, 7

fia_load(), 5, 8

fia_split_composite_ids, 7

fia_split_composite_ids(), 2

fia_tidy, 8

fia_tidy(), 4

interpolate_data(), 4

rFIA::readFIA(), 7

state_areas, 9