Package 'FIAstemmap'

September 24, 2026

Title Tree Canopy Modeling for USDA Forest Inventory and Analysis Plots

Version 2.0.0

Description Maps individual tree stem locations on field plots of the Forest Inventory and Analysis Program of USDA Forest Service (<https://research.fs.usda.gov/programs/nfi>). Stem locations are mapped in cartesian coordinate space based on field-measured distance and azimuth from subplot and microplot centers. Per-tree crown widths are estimated using a curated set of allometric equations with coverage for the conterminous US. Spatial descriptors of tree point pattern are computed at the whole plot level. Several stand height metrics are also computed and provided in the output. The spatial representation of modeled tree crowns is used to generate estimates of fractional tree canopy cover at the microplot, subplot and whole plot levels. Convenience functions are provided for efficient data processing. Exploratory data analysis is also facilitated via integration with the 'spatstat' packages.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Imports bit64, cli, gdalraster (>= 2.5.0), grDevices, methods, spatstat.explore, spatstat.geom, stats

Suggests glue (>= 1.6.0), spelling, testthat (>= 3.0.0), vctrs (>= 0.3.0)

URL https://firelab.github.io/FIAstemmap/, https://github.com/firelab/FIAstemmap

BugReports https://github.com/firelab/FIAstemmap/issues

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Language en-US

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Chris Toney [aut, cre] (ORCID: <https://orcid.org/0000-0001-5734-6510>)

Maintainer Chris Toney <jctoney@gmail.com>

Repository CRAN

Date/Publication 2026-09-24 14:00:15 UTC

Contents

`calc_crown_overlay` *Compute fractional tree canopy cover of a subplot/microplot by crown overlay*

Description

`calc_crown_overlay()` computes the proportion of a circular polygon covered by a given set of tree crowns modeled as discs and having spatially explicit stem locations. The sampled area is generally an FIA subplot with radius 24 ft (7.315 m) for trees with diameter `>= 5 in.` (12.7 cm), or an FIA microplot with radius 6.8 ft (2.073 m) for trees `>= 1 in.` (2.54 cm) but `< 5 in.` (12.7 cm) diameter (denoted as "saplings"). Stem locations are specified as distance and azimuth from subplot/microplot center.

Usage

```
calc_crown_overlay(tree_list, sample_radius, digits = 1)
```

Arguments

`tree_list`	A data frame containing tree records for a subplot/microplot. Must have columns `DIST` (stem distance from subplot/microplot center in the same units as `sample_radius`), `AZIMUTH` (horizontal angle from subplot/microplot center to the stem location, in the range `0:359`) and `CRWIDTH` (tree crown width in the same units as `sample_radius` and `DIST`).
`sample_radius`	A numeric value giving the radius of the circular subplot/microplot.
`digits`	Optional integer indicating the number of digits to keep in the return value (defaults to `1`, will be passed to `round()`).

Value

Estimated tree canopy cover as percent of the area specified by `sample_radius` that is covered by a vertical projection of circular crowns.

Note

This function does not perform any filtering based on `SUBP` (subplot), `STATUSCD`(live vs dead trees) or`DIA` (mature trees vs saplings). The input tree list is assumed to be filtered to the specific set of live trees for one subplot or microplot with the given `sample_radius`.

See Also

`calc_tcc_metrics()`

Examples

```
# subplot 1 of the `plantation` plot
trees <- within(plantation, CRWIDTH <- calc_crwidth(plantation))
trees[trees$SUBP == 1 & trees$DIA >= 5, ] |>
  calc_crown_overlay(sample_radius = 24)

plot_crowns(trees, subplot = 1, main = "plantation subplot 1")
```

`calc_crwidth` *Predict individual tree crown width using species-specific equations*

Description

`calc_crwidth()` predicts tree crown width using species-specific regression equations from the literature.

Usage

```
calc_crwidth(tree_table, digits = 1)
```

Arguments

`tree_table`	A data frame containing tree records. Must have columns `SPCD` (FIA integer species code), `STATUSCD` (FIA integer tree status code, 1 = live) and `DIA` (FIA tree diameter in inches).
`digits`	Optional integer indicating the number of digits to keep in the return values (defaults to `1`).

Details

Crown width is predicted from tree diameter using coefficients provided in the lookup table cw_coef (see `?cw_coef`) The method also incorporates adjustment factors used to derive crown width estimates for FIA "saplings", i.e., trees less than 5.0 in. (12.7 cm) diameter but greater than or equal to 1.0 in. (2.54 cm) diameter. Details are described in the documentation for the lookup table cw_sapling_adj.

Large diameter trees in the temperate rain forests of the Pacific Northwest region can far exceed the range of diameters in the broadly applicable datasets that have been used to develop crown width prediction equations (Bechtold 2003, 2004). To avoid extrapolation beyond the range of the model fitting data in those cases, `calc_crwidth()` makes use of the "old growth" equation presented by Gill et al. (2000) to estimate crown width for nine tree species when their diameter is greater than 50 in. (127 cm).

Value

A numeric vector of length `nrow(tree_table)` with predicted crown width in feet for live trees. `NA` is returned for trees with `STATUSCD != 1`.

References

Bechtold, W.A. 2003. Crown-diameter prediction models for 87 species of stand-grown trees in the eastern United States. *Southern Journal of Applied Forestry*, 27(4): 269-278.

Bechtold, W.A. 2004. Largest-crown-width prediction models for 53 species in the western United States. *Western Journal of Applied Forestry*, 19(4): 245-251.

Gill, S.J., G.S. Biging, E.C. Murphy. 2000. Modeling conifer tree crown radius and estimating canopy cover. *Forest Ecology and Management*, 126(3): 405-416.

See Also

cw_coef, cw_sapling_adj

Examples

```
calc_crwidth(plantation)
```

calc_ht_metrics *Calculate stand height metrics from tree list data*

Description

These functions compute several stand height metrics from tree list data.

Usage

```
calc_ht_metrics(tree_list, digits = 1)

calc_landfire_stand_ht(
  subp_overlay_mean,
  micr_overlay_mean,
  numTrees,
  meanTreeHtDomBAW,
  meanTreeHtBAW,
  meanSapHt
)
```

Arguments

tree_list	A data frame with tree records for one FIA plot. Must have columns DIA (tree diameter), HT (tree height), ACTUALHT (tree actual height, ACTUALHT < HT indicating a broken top), CCLCD (FIA crown class code), TPA_UNADJ (trees per acre).
digits	Optional integer indicating the number of digits to keep in the return values (defaults to 1).
subp_overlay_mean	A numeric vector, value(s) of subp_overlay_mean from the output of calc_tcc_metrics().
micr_overlay_mean	A numeric vector, value(s) of micr_overlay_mean from the output of calc_tcc_metrics().
numTrees	A numeric vector, value(s) of numTrees from the output of calc_ht_metrics().
meanTreeHtDomBAW	A numeric vector, value(s) of meanTreeHtDomBAW from the output of calc_ht_metrics().
meanTreeHtBAW	A numeric vector, value(s) of meanTreeHtBAW from the output of calc_ht_metrics().
meanSapHt	A numeric vector, value(s) of meanTreeHmeanSapHttBAW from the output of calc_ht_metrics().

Details

calc_ht_metrics() computes several stand height metrics for a given tree list. The return value is a named list as described below.

calc_landfire_stand_ht() computes LANDFIRE stand height. This metric is computed separately since it depends on canopy cover estimates for the sapling and overstory layers derived by overlaying modeled crowns. The input data are given as vectors of values for one or more plots (all

input vectors must have the same length). The return value is a numeric vector of stand heights, with length equal to the number of elements in each of the input vectors.

Stand height metrics are based on live trees (`STATUSCD == 1`), and are are assigned `0` by definition if no live trees are present. `calc_ht_metrics()`returns a named list with the following elements:

- `$numTrees`: number of live trees `>= 5.0` in. (`12.7` cm) diameter
- `$meanTreeHt`: mean height of trees `>= 5.0` in. (`12.7` cm) diameter
- `$meanTreeHtBAW`: basal-area weighted mean height of trees `>= 5.0` in. (`12.7` cm) diameter
- `$meanTreeHtDom`: mean height of canopy dominant/co-dominant trees `>= 5.0` in. (`12.7` cm) diameter
- `$meanTreeHtDomBAW`: basal-area weighted mean height of canopy dominant/co-dominant trees `>= 5.0` in. (`12.7` cm) diameter
- `$maxTreeHt`: height of the tallest tree `>= 5.0` in. (`12.7` cm) diameter
- `$predomTreeHt`: predominant tree height, as the mean height of the tallest trees `>= 5.0` in. (`12.7` cm) diameter comprising up to `16` trees per acre (`39.5` trees per hectare)
- `$numSaplings`: number of live saplings (trees `>= 1.0` in. but `< 5.0` in. diameter, i.e., `>= 2.54` cm but `< 12.7` cm)
- `$meanSapHt`: mean height of saplings
- `$maxSapHt`: height of the tallest sapling

For the purpose of height calculations, metrics based on "canopy dominant/co-dominant" include open grown trees, i.e., include trees with FIA crown class codes `CCLCD` of `1` (open grown), `2` (dominant) or `3` (co-dominant), but exclude trees with `CCLCD` of `4` (intermediate) or `5` (over-topped).

LANDFIRE stand height is a metric computed for FIA plots used as reference data supporting development of the Existing Vegetation Height (EVH) raster product (https://www.landfire.gov/vegetation/evh). It is generally the basal-area weighted mean height of canopy dominant/co-dominant trees (i.e., `meanTreeHtDomBAW`). However, this metric attempts to identify plots that may be best characterized as sapling stage, in which case stand height is the mean height of saplings (`meanSapHt`). Sapling-stage plots are defined using arbitrary thresholds of canopy cover, estimated separately for the sapling layer based on microplot data, and the overstory tree layer based on subplot measurements. See `calc_tcc_metrics()` for variable definitions. A plot is considered sapling stage if `subp_overlay_mean <= 10` and `micr_overlay_mean >= 3 * subp_overlay_mean`.

Value

`calc_ht_metrics()` returns a named list of stand height metrics as described in Details. `calc_landfire_stand_ht()` returns a numeric vector of stand heights, with length equal to the number of elements in each of the input vectors.

Examples

```
calc_ht_metrics(plantation)

calc_landfire_stand_ht(86, 11, 89, 45, 45, 34)
```

`calc_tcc_metrics` *Predict plot-level canopy cover from individual tree measurements*

Description

`calc_tcc_metrics()` computes predicted plot-level tree canopy cover (TCC) from standard field inventory measurements. By default, the output includes a full set of stand structure variables used to model the plot-level TCC value (see Details).

Usage

```
calc_tcc_metrics(
  tree_list,
  stem_map = TRUE,
  full_output = TRUE,
  digits = 1,
  ...
)
```

Arguments

`tree_list`	A data frame with tree records for one FIA plot. In general, the input data frame will have the columns specified in DEFAULT_TREE_COLUMNS (see `?DEFAULT_TREE_COLUMNS`). Potentially, only a subset of those columns will be needed depending on values given for the arguments `stem_map` and `full_output` described below. If the input data frame has a column named `"CRWIDTH"` it will be used for tree crown width values, otherwise, crown widths will be calculated with a call to `calc_crwidth()`.
`stem_map`	A logical value indicating whether to map individual tree stems explicitly, using coordinates specified in terms of distance and azimuth from subplot/microplot centers. The default is `TRUE`, in which case the input `tree_list` must contain columns `"DIST"` and `"AZIMUTH"`. This argument may be set to `FALSE` if individual tree locations are not available, in which case TCC will be predicted assuming a random arrangement of tree locations (see Details).
`full_output`	A logical value indicating whether to include the full set of components used to derive the plot-level prediction. By default, the output list includes subplot-level TCC estimates, live tree and sapling counts, stand height metrics, and point pattern statistics, depending on the value given for `stem_map` (see Details).
`digits`	Optional integer indicating the number of digits to keep in the return values (defaults to `1`). May be passed to `calc_crwidth()` and `calc_ht_metrics()`.
`...`	Optional arguments passed to `create_fia_ppp()` if `stem_map = TRUE`.

Details

This function provides two methods for predicting plot-level TCC.

The default "stem-map" method requires individual tree coordinates to be given in the input as distance and azimuth from subplot centers for trees with diameter >= 5 in. (12.7 cm), and from microplot centers for "saplings" having diameter >= 1 in. (2.54 cm) but < 5 in. (12.7 cm). This method involves mapping trees spatially within the plot boundary to account for crown overlap explicitly, along with empirical modeling of the understory sapling contribution to total canopy cover (Toney et al. 2009). The empirical model for the sapling component also uses the spatial point pattern of overstory trees as a predictor variable (using a square root transformation of Ripley's edge-corrected K-function, Ripley 1977, Stoyan and Penttinen 2000).

Alternatively, TCC can be predicted using a simplified approach that does not include exact stem placement within the plot boundary (`stem_map = FALSE`). A random arrangement of stems is assumed in that case. This is the method used to estimate tree canopy cover in the Forest Vegetation Simulator (Crookston and Stage 1999).

Both methods require estimates of individual tree crown widths, which are computed with `calc_crwidth()` if not provided in the input tree list.

The stem-map method also requires computation of several stand structure metrics which are used in various components of the overall model used to derive a plot-level TCC estimate. These additional variables include:

- individual subplot and microplot crown overlays via `calc_crown_overlay()`
- a stand height metric (`meanTreeHtBAW`), and plot-level counts of mature trees and saplings, via `calc_ht_metrics()`
- descriptive spatial statistics for the overstory tree point pattern via `create_fia_ppp() |> spatstat.explore::Lest()`

By default, `calc_tcc_metrics()` returns a named list containing the plot-level modeled TCC value, along with those additional component variables. Specific elements of the returned list include some or all of the following, conditionally:

- `$model_tcc`: plot-level predicted canopy cover of trees >= 1 inch (2.54 cm) diameter, derived by one of the two methods described above depending on the value given for argument `stem_map = TRUE|FALSE`

If the stem-map method is used, then TCC values derived by crown overlay on the individual subplot and microplot boundaries are included, along with means of the four subplot/microplot values:

- `$subpN_crown_overlay`: estimated canopy cover of trees >= 5-in. (12.7 cm) diameter in subplot `N` based on crown overlay (`N = 1:4`)
- `$subp_overlay_mean`: mean of the four subplot crown overlays
- `$micrN_crown_overlay`: estimated canopy cover of saplings in the microplot of subplot `N` based on crown overlay (`N = 1:4`)
- `$micr_overlay_mean`: mean of the four microplot crown overlays

A set of spatial point pattern statistics is also included when the stem-map method is used. A square root transformation of Ripley's K function using isotropic edge correction is computed with `spatstat.explore::Lest()` for trees >= 5-in. (12.7 cm) diameter within the four-subplot observation window. The mean of the following values is a predictor variable in a linear regression model used to estimate the sapling contribution to total tree canopy cover:

- `$L_rft`: estimates of the L-function at `r` feet (`r = 6`, `8`, `10`, and `12`)

If the argument `full_output = TRUE` (the default), then the output will also include all of the the named elements from the output of `calc_ht_metrics()`.

Value

If `full_output = TRUE`, a named list with element `model_tcc` containing the plot-level predicted tree canopy cover as percent (`0:100`), and additional named elements containing stand structure metrics as described in Details. If `full_output = FALSE`, a single numeric value of plot-level predicted TCC is returned instead.

References

Crookston, N.L. and A.R. Stage. (1999). Percent canopy cover and stand structure statistics from the Forest Vegetation Simulator. Gen. Tech. Rep. RMRS-GTR-24. Ogden, UT: U. S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 11 p. https://research.fs.usda.gov/treesearch/6261.

Ripley, B.D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(2): 172–192. doi:10.1111/j.25176161.1977.tb01615.x.

Stoyan, D., and Penttinen, A. (2000). Recent applications of point process methods in forestry statistics. *Statistical Science*, 15(1), 61–78. http://www.jstor.org/stable/2676677.

Toney, C., J.D. Shaw and M.D. Nelson. 2009. A stem-map model for predicting tree canopy cover of Forest Inventory and Analysis (FIA) plots. In: McWilliams, Will; Moisen, Gretchen; Czaplewski, Ray, comps. *Forest Inventory and Analysis (FIA) Symposium 2008*; October 21-23, 2008; Park City, UT. Proc. RMRS-P-56CD. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 19 p. https://research.fs.usda.gov/treesearch/33381.

See Also

`calc_crwidth()`, `calc_crown_overlay()`, `calc_ht_metrics()`, `create_fia_ppp()`

Examples

```
# using the spatially explicit "stem-map model" by default
calc_tcc_metrics(plantation)

# return only the predicted TCC value (`$model_tcc`)
calc_tcc_metrics(plantation, full_output = FALSE)

# using the "FVS method" which assumes random tree locations
calc_tcc_metrics(plantation, stem_map = FALSE, full_output = FALSE)
```

---

convert_units *Convenience functions for common unit conversions*

Description

Functions to convert between US customary units and SI units for measurements of length and area commonly used with tree data.

Usage

```
ft_to_m(x)

m_to_ft(x)

in_to_cm(x)

cm_to_in(x)

ac_to_ha(x)

ha_to_ac(x)
```

Arguments

x	Numeric vector of values to convert.

Details

`ft_to_m(x)` converts feet to meters.

`m_to_ft(x)` converts meters to feet.

`in_to_cm(x)` converts inches to centimeters.

`cm_to_in(x)` converts centimeters to inches.

`ac_to_ha()` converts acres to hectares.

`ha_to_ac()` converts hectares to acres.

Value

A numeric vector of the converted values.

Note

The hectare (ha) is technically a non-SI unit of area that is accepted for use with SI.

Examples

```
ft_to_m(1)

m_to_ft(1)

in_to_cm(1)

cm_to_in(1)

ac_to_ha(1)

ha_to_ac(1)
```

cw_coef *Regression coefficients for predicting tree crown width*

Description

A curated set of linear regression coefficients for predicting crown width from stem diameter of tree species in the conterminous US.

Usage

```
cw_coef
```

Format

`cw_coef`:

A data frame with 430 rows and 8 columns:

`symbol` `character`, USDA PLANTS Database species symbol

`SPCD` `integer`, FIA tree species code or `-1`

`common_name` `character`, FIA tree species common name

`surrogate` `character`, Common name of surrogate species if applicable

`b0` `numeric`, Regression b0 coefficient

`b1` `numeric`, Regression b1 coefficient

`b2` `numeric`, Regression b2 coefficient

`reference` `character`, Literature source of the species coefficients (see References)

Details

The regression equation is of the general form:

```
CW = b0 + b1 * DIA + b2 * DIA^2
```

where `CW` is the predicted tree crown diameter in feet, `DIA` is FIA stem diameter in inches, and `b0`, `b1`, `b2` are the regression coefficients. The quadratic term `b2` is not included in the regression models for some species, and has been assigned `0` in that case for purposes of this lookup table.

In cases that species-specific equations were not available in the literature, surrogate species were assigned based on subjectively similar tree physiognomy.

Source

Toney et al. 2009. A stem-map model for predicting tree canopy cover of Forest Inventory and Analysis (FIA) plots. https://research.fs.usda.gov/treesearch/33381.

References

Bechtold, W.A. 2003. Crown-diameter prediction models for 87 species of stand-grown trees in the eastern United States. *Southern Journal of Applied Forestry*, 27(4): 269-278.

Bechtold, W.A. 2004. Largest-crown-width prediction models for 53 species in the western United States. *Western Journal of Applied Forestry*, 19(4): 245-251.

Gill, S.J., G.S. Biging, E.C. Murphy. 2000. Modeling conifer tree crown radius and estimating canopy cover. *Forest Ecology and Management*, 126(3): 405-416.

Examples

```
cw_coef[cw_coef$SPCD == 17, ]
```

---

cw_sapling_adj *Sapling crown width adjustment factors*

Description

A species-specific lookup table of estimated adjustment factors for crown width (CW) of saplings based on data from Bragg (2001).

Usage

```
cw_sapling_adj
```

Format

`cw_sapling_adj`**:**

A data frame with 23 rows and 5 columns:

`SPCD` `integer`, FIA tree species code

`adj_1inch` `numeric`, CW adjustment factor at 1 in. DIA relative to 5 in. DIA

`adj_2inch` `numeric`, CW adjustment factor at 2 in. DIA relative to 5 in. DIA

`adj_3inch` `numeric`, CW adjustment factor at 3 in. DIA relative to 5 in. DIA

`adj_4inch` `numeric`, CW adjustment factor at 4 in. DIA relative to 5 in. DIA

Details

FIA "saplings" are trees less than 5.0 in. (12.7 cm) diameter but greater than or equal to 1.0 in. (2.54 cm) diameter. In general, the data available to fit regression models predicting crown width (e.g., Bechtold 2003, 2004, see cw_coef) do not include trees with diameter less than 5.0 in. (12.7 cm). Extrapolating beyond the range of the model fitting data is undesirable, especially since a quadratic term is used in the regression equations for some species.

Adjustment is based on the proportion of crown width predicted for 5-in. (12.7 cm) diameter, at each 1-in. (2.54 cm) increment below that. Intermediate values are interpolated in the crown width prediction method. Mean adjustment factors are used if a species-specific adjustment is not available.

Source

Toney et al. 2009. A stem-map model for predicting tree canopy cover of Forest Inventory and Analysis (FIA) plots. https://research.fs.usda.gov/treesearch/33381.

References

Bragg, D.C. 2001. A local basal area adjustment for crown width prediction. *Northern Journal of Applied Forestry* 18(1):22-28.

Examples

```
cw_coef[cw_coef$SPCD == 261, ]

cw_sapling_adj[cw_sapling_adj$SPCD == 261, ]
```

DEFAULT_TREE_COLUMNS *Names of the FIADB TREE columns used by default in* **FIAstemmap**

Description

DEFAULT_TREE_COLUMNS is a character vector of column names in the FIADB TREE table that are used to compute various tree- and plot-level derived variables in **FIAstemmap**. Note that certain outputs can be generated without this full set of TREE attributes, but these are the ones generally needed for all functionality in the package.

Usage

```
DEFAULT_TREE_COLUMNS
```

Value

A character vector of 12 strings containing the names of columns in the FIADB TREE table used by default in **FIAstemmap**.

Source

https://research.fs.usda.gov/products/dataandtools/fia-datamart

Examples

```
DEFAULT_TREE_COLUMNS
```

load_tree_data *Load tree data from a file or database connection*

Description

load_tree_data() fetches tree records from a data source, typically a comma-separated values (CSV) file, a SQLite database file (.db, .sqlite, .gpkg), or a PostgreSQL database connection. Other data sources are also possible. File-based sources can be read from compressed archives without prior extraction if desired (e.g., .zip), and network-hosted files can be read directly without prior download (see Details).

Usage

```
load_tree_data(
  src,
  table = NULL,
  columns = DEFAULT_TREE_COLUMNS,
  sql = NULL,
  quoted_cols_as_char = TRUE,
  colnames_toupper = TRUE
)
```

Arguments

src	A character string specifying the data source as a file name or database connection string (see Details).
table	Optional character string giving the name of a table in src from which tree records will be fetched. Generally needed with database sources containing multiple tables (as opposed to a single-table source such as a CSV file).
columns	Optional character vector specifying a subset of column names in the source table to include in the result set. Defaults to DEFAULT_TREE_COLUMNS. Can also be set to NULL or empty string ("") to read all columns in the source table.
sql	Optional character string containing a SQL SELECT statement to execute on src (instead of selecting all records, potentially from a subset of columns, i.e., mutually exclusive with table and/or columns).
quoted_cols_as_char	A logical value indicating whether to auto-detect columns that contain quoted values as "character" type, TRUE by default. Only used when src is a CSV file.
colnames_toupper	Logical value, TRUE to enforce upper case column names in the returned data frame (the default).

Details

A data source is most commonly specified as one of the following:

CSV file
Path to a text file with ".csv" extension. For files structured as CSV, but not ending with the ".csv" extension, a "CSV:" prefix can be added before the filename to force loading as CSV format.

SQLite database
Path to a SQLite file. The file extension is generally either ".db" or ".sqlite", but GeoPackage files with the ".gpkg" extension are also supported.

PostgreSQL database
A connection string in one of the following formats:

```
"PG:dbname=databasename"

"PG:dbname='db' host='addr' port='5432' user='x' password='y'"

"PG:service=servicename"

"postgresql://[user[:pwd]@][netloc][:port][/dbname][?param1=val1&...]"
```

GDAL Virtual File Systems are also supported. This allows, for example, reading from compressed archives such as ".zip" without prior extraction. The syntax in that case uses the "/vsizip/" prefix:

```
# relative path to the .zip:
f <- "/vsizip/MT_CSV.zip/MT_TREE.csv"

# absolute path to the .zip:
f <- "/vsizip//home/ctoney/data/MT_CSV.zip/MT_TREE.csv"

# on Windows:
f <- "/vsizip/c:/users/ctoney/MT_CSV.zip/MT_TREE.csv"
```

Network-hosted files can also be read without prior download using the "/vsicurl/" prefix:

```
f <- "/vsicurl/https://apps.fs.usda.gov/fia/datamart/CSV/MT_TREE.csv"
```

For more details, including supported VSI prefixes for cloud storage services and other virtual file systems, see https://gdal.org/en/stable/user/virtual_file_systems.html.

Value

A data frame containing tree records fetched from src.

Note

src can be any GDAL supported dataset. A full list of formats supported by the current GDAL installation can be obtained with:

```
fmt <- gdalraster::gdal_formats()
fmt$long_name[fmt$vector]
```

For more details: https://gdal.org/en/stable/drivers/vector/index.html

load_tree_data() from a PostgreSQL database requires GDAL built with support for the PostgreSQL client library (can be checked with gdalraster::gdal_formats("postgresql")).

Column names are case-sensitive in **FIAstemmap** functions, and are assumed to follow the FIADB upper case naming convention.

If column PLT_CN is present, it will be read or coerced if necessary to R "character" type consistent with its data type in FIADB (i.e., a string but all digits).

See Also

DEFAULT_TREE_COLUMNS,process_tree_data()

Examples

```
# Lolo NF, single-condition forest plots, INVYR 2022, from public FIADB
f <- system.file("extdata/mt_lnf_2022_1cond_tree.csv", package="FIAstemmap")
tree <- load_tree_data(f)

head(tree)
```

---

plantation *Tree list for a pine plantation*

Description

An example tree list for an FIA plot in a loblolly pine plantation.

Usage

```
plantation
```

Format

plantation:

A data frame with 91 rows and 12 columns:

PLT_CN character, Plot unique identifier

SUBP integer, Subplot number

TREE integer, Tree number

AZIMUTH integer, Horizontal angle from subplot center to the stem location

DIST numeric, Distance in feet from subplot center to the stem location

STATUSCD integer, Tree status code: 1 = live, 2 = standing dead

SPCD integer, FIA tree species code

DIA numeric, Tree diameter at breast height in inches

HT numeric, Tree height in feet

ACTUALHT numeric, Actual height in ft (ACTUALHT < HT indicates a broken top)

CCLCD integer, Tree crown class code

TPA_UNADJ numeric, Trees per acre expansion factor

Note

A synthetic plot unique identifier is used in example tree list datasets.

Source

https://research.fs.usda.gov/programs/nfi

Examples

```
plot_crowns(plantation, main = "plantation plot")
```

---

plot_crowns *Display modeled tree crowns projected vertically on subplot boundaries*

Description

`plot_crowns()` draws vertically projected tree crowns as discs overlaid on FIA subplot or microplot boundaries. The full four-subplot cluster, or individual subplots, can be displayed with trees `>= 5.0` in. (`12.7` cm) diameter. Individual microplots can also be displayed with saplings (i.e., trees `< 5` in. diameter).

Usage

```
plot_crowns(
  tree_list,
  subplot = NULL,
  microplot = FALSE,
  linear_unit = "ft",
  main = "",
  crown_col = "#328e13",
  stem_col = "#b85e00",
  subp_border_lwd = 3,
  subp_border_col = "gray61"
)
```

Arguments

tree_list	A data frame with tree records for one FIA plot. Must have columns SUBP (FIA subplot number), STATUSCD (FIA integer tree status, 1 = live), DIA (tree diameter), HT (tree height), ACTUALHT (tree actual height, ACTUALHT < HT indicating a broken top), DIST (stem distance from subplot/microplot center) and AZIMUTH (horizontal angle from subplot/microplot center to the stem location, in range 0:359). If the input data frame has a column named "CRWIDTH" it will be used for tree crown width values, otherwise, crown widths will be calculated with a call to calc_crwidth().
subplot	Optional integer subplot number in the range 1:4 indicating a specific subplot for display. May be NULL or NA to display the whole four-point cluster plot.
microplot	A logical value, TRUE to display the modeled crowns of saplings overlaid of the microplot boundary of subplot = n. The default is FALSE. Ignored if subplot is not specified.
linear_unit	An optional character string specifying the linear distance unit. Defaults to the native FIA unit of "ft", but may be set to "m" instead (or "meter" / "metre"), in which case subplot boundaries will be displayed in meters, tree distances and crown widths are assumed to be given in meters, and tree diameters are assumed to be given in centimeters.
main	Character string giving the main plot title (on top).
crown_col	The color of tree crowns, e.g., either a color name (as listed by colors()) or a hexadecimal string.
stem_col	The color of tree stems when plotting an individual subplot or microplot (see crown_col above).
subp_border_lwd	The line width of subplot boundaries. Must a positive number.
subp_border_col	The color of subplot boundaries (see crown_col above).

Value

The input, invisibly.

See Also

calc_crwidth(), calc_crown_overlay()

Examples

```
plot_crowns(plantation, main = "plantation plot")

plot_crowns(plantation, subplot = 4, main = "plantation subplot 4")

plot_crowns(plantation, subplot = 4, microplot = TRUE,
            main = "plantation microplot 4")

# using SI units
```

```
metric_trees <- within(plantation, {
  CRWIDTH <- calc_crwidth(plantation) |> ft_to_m()
  rm(DIST, DIA)
  DIST <- ft_to_m(plantation$DIST)
  DIA <- in_to_cm(plantation$DIA)
})
plot_crowns(metric_trees, linear_unit = "meter",
            main = "plantation plot (SI units)")
```

`process_tree_data` *Generate plot-level stand structure metrics for a tree dataset*

Description

`process_tree_data()` takes a table of tree records for a set of forest inventory plots as input, and generates selected plot-level stand structure metrics.

Usage

```
process_tree_data(tree_table, stem_map = TRUE, full_output = TRUE, digits = 1)
```

Arguments

`tree_table`	A data frame containing tree records for a set of forest inventory plots. Must have column `PLT_CN` containing the plot unique identifier for each tree. Other required columns are those of `calc_crwidth()` (if column `CRWIDTH` is not included), `calc_ht_metrics()` and `calc_tcc_metrics()`, depending on values given for `stem_map` and `full_output`.
`stem_map`	A logical value indicating whether to map individual tree stems explicitly, using coordinates specified in terms of distance and azimuth from subplot/microplot centers. The default is `TRUE`, in which case the input `tree_table` must contain columns `"DIST"` and `"AZIMUTH"`. This argument may be set to `FALSE` if individual tree locations are not available, in which case TCC will be predicted assuming a random arrangement of tree locations (see Details for `calc_tcc_metrics()`).
`full_output`	A logical value indicating whether to include the full set of components used to derive the plot-level TCC prediction. By default, the output data includes subplot-level TCC estimates, live tree and sapling counts, stand height metrics, and point pattern statistics, depending on the value given for `stem_map` (see Details for `calc_tcc_metrics()`).
`digits`	Optional integer indicating the number of digits to keep in the return values (defaults to `1`). May be passed to `calc_crwidth()` and `calc_ht_metrics()`.

Value

A data frame with one row for each unique `PLT_CN` in the input `tree_table`, and additional columns containing the output of `calc_tcc_metrics()` conditional on the values given for `stem_map` and `full_output`.

See Also

calc_ht_metrics(), calc_tcc_metrics(), load_tree_data()

Examples

```
# Lolo NF, single-condition forest plots, INVYR 2022, from public FIADB
f <- system.file("extdata/mt_lnf_2022_1cond_tree.csv", package="FIAstemmap")
tree_table <- load_tree_data(f)

process_tree_data(tree_table, stem_map = FALSE, full_output = TRUE)
```

---

spatstat_helpers *Analyze the spatial pattern of trees on an FIA plot*

Description

These functions facilitate point pattern analysis of FIA tree data using the **spatstat** library.

Usage

```
create_fia_owin(linear_unit = "ft", macroplot = FALSE, npoly = 360)

create_fia_ppp(
  tree_list,
  live_trees = TRUE,
  min_dia = 5,
  linear_unit = "ft",
  macroplot = FALSE,
  window = NULL,
  mark_cols = NULL,
  mark_as_factor = NULL
)
```

Arguments

linear_unit	An optional character string specifying the linear distance unit. Defaults to the native FIA unit of "ft", but may be set to "m" instead (or "meter" / "metre").
macroplot	An optional logical value. The default is FALSE, which defines the FIA plot footprint in terms of the standard four-subplot configuration with subplot radius of 24 ft (7.3152 m). By default, "macroplot trees" having DIST outside the subplot boundary are not included. This argument may be set to TRUE in which case the observation window will be defined using the FIA optional "macroplot" configuration instead (58.9 ft or 18.227 m radius, used only in certain areas of the Pacific Northwest FIA region).
npoly	Integer value giving the number of edges to use for polygon approximation. Defaults to 360.

tree_list	A data frame containing a set of tree records for one FIA plot (see Details).
live_trees	A logical value, TRUE to include live trees only (the default, i.e., STATUSCD == 1).
min_dia	A numeric value specifying the minimum diameter threshold for included trees. The default is 5.0. Trees less than 5-in. diameter but greater than or equal to 1.0-in. diameter, denoted as "saplings", are only recorded in FIA microplots so cannot be stem-mapped across the full 4-subplot footprint.
window	An optional object of class "owin" defining the observation window of an FIA plot in the 2-D plane. Defaults to create_fia_owin(linear_unit, macroplot).
mark_cols	An optional character vector of column names in tree_list to designate as **spatstat** marks which carry additional information for each data point in a point pattern object.
mark_as_factor	An optional subset of mark_cols to be treated as factor marks. If not already factor, these will be coerced as such upon input. factor marks are those that take only a finite number of possible values (e.g. colors or types).

Details

create_fia_owin() returns an object of class "owin" from the **spatstat** library. This object represents the generic 2-D observation window for the nationally standard FIA plot design which is a four-point cluster of circular subplots. Used when creating a **spatstat** point pattern object for an FIA tree list.

create_fia_ppp() returns an object of class "ppp" representing the point pattern of an FIA tree list in the 2-D plane. A point pattern object defines the dataset for using a stem-mapped FIA plot with functions of the **spatstat** library.

The standard set of columns for tree list data in **FIAstemmap** is given below, along with the status of each column as required or optional for create_fia_ppp():

- PLT_CN: optional, create_fia_ppp() assumes input is for one plot
- SUBP: required subplot number
- TREE: required tree number within a subplot
- AZIMUTH: required horizontal angle from subplot center
- DIST: required distance from subplot center
- STATUSCD: required tree status code (1 = live, 2 = dead)
- SPCD: optional FIA species code
- DIA: optional tree diameter
- HT: optional tree height
- ACTUALHT: optional tree actual height (accounts for broken top)
- CCLCD: optional crown class code
- TPA_UNADJ: optional tree expansion factor (per acre)
- CRWIDTH: optional crown width (see calc_crwidth())

Value

`create_fia_owin()` returns an object of class `"owin"` from the **spatstat** library representing the generic 2-D observation window for the nationally standard FIA plot design. `create_fia_ppp()` returns an object of class `"ppp"` from **spatstat** representing the point pattern of an FIA tree list in the 2-D plane.

References

Baddeley, A., E. Rubak, R. Turner. 2015. *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC Press, London. ISBN 9781482210200, https://www.routledge.com/Spatial-Point-Patterns-Methodology-and-Applications-with-R/Baddeley-Rubak-Turner/p/book/9781482210200/.

Ripley, B.D. 1977. Modelling spatial patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(2): 172–192. doi:10.1111/j.25176161.1977.tb01615.x.

See Also

spatstat package: https://cran.r-project.org/package=spatstat

Examples

```
## observation window for the standard FIA plot design
w <- create_fia_owin()
summary(w)

# or using metric units
w <- create_fia_owin("m")
summary(w)

plot(w, main = "FIA standard four-subplot design")

## point pattern object for the `plantation` example data
X <- create_fia_ppp(plantation)
summary(X)

plot(X, pch = 16, main = "Loblolly pine plantation")

# plot trees as trees :)
X <- create_fia_ppp(plantation, mark_cols = "SPCD")
plot(X, shape = "arrows", direction = 90, size = 12, cols = "darkgreen",
     legend = FALSE, main = "Loblolly pine plantation")

# Ripley's K-function
K <- spatstat.explore::Kest(X, rmax = 12, correction = "isotropic")
plot(K, main = "Ripley's K for the plantation FIA plot")

## point pattern object for the `western_redcedar` example data
## tree distances may be specified in meters
trees <- within(western_redcedar, DIST <- ft_to_m(DIST))

X <- create_fia_ppp(trees, linear_unit = "m")
```

```
summary(X)

plot(X, pch = 16, main = "Western redcedar FIA plot")

# Ripley's K-function
K <- spatstat.explore::Kest(X, rmax = ft_to_m(12), correction = "isotropic")
plot(K, main = "Ripley's K for the western redcedar FIA plot")
```

`western_redcedar` *Tree list for a western redcedar forest*

Description

An example tree list for an FIA plot in a western redcedar forest.

Usage

```
western_redcedar
```

Format

`western_redcedar`**:**
A data frame with 33 rows and 12 columns:

`PLT_CN` `character`, Plot unique identifier
`SUBP` `integer`, Subplot number
`TREE` `integer`, Tree number
`AZIMUTH` `integer`, Horizontal angle from subplot center to the stem location
`DIST` `numeric`, Distance in feet from subplot center to the stem location
`STATUSCD` `integer`, Tree status code: 1 = live, 2 = standing dead
`SPCD` `integer`, FIA tree species code
`DIA` `numeric`, Tree diameter at breast height in inches
`HT` `numeric`, Tree height in feet
`ACTUALHT` `numeric`, Actual height in ft (ACTUALHT < HT indicates a broken top)
`CCLCD` `integer`, Tree crown class code
`TPA_UNADJ` `numeric`, Trees per acre expansion factor

Note

A synthetic plot unique identifier is used in example tree list datasets.

Source

https://research.fs.usda.gov/programs/nfi

Examples

```
plot_crowns(western_redcedar, main = "western redcedar plot")
```

Index