Package 'fgeo.plot'

September 3, 2022

Title Plot ForestGEO Data

Version 1.1.11

Description To help you access, transform, analyze, and visualize ForestGEO data, we developed a collection of R packages (<https://forestgeo.github.io/fgeo/>). This package, in particular, helps you to plot ForestGEO data. To learn more about ForestGEO visit <https://forestgeo.si.edu/>.

License GPL-3

URL https://github.com/forestgeo/fgeo.plot, https://forestgeo.github.io/fgeo.plot/

BugReports https://github.com/forestgeo/fgeo.plot/issues

Depends R (>= 3.3)

Imports dplyr (>= 0.8.0.1), fgeo.tool (>= 1.2.4), ggplot2 (>= 3.1.1), ggrepel (>= 0.8.1), glue (>= 1.3.1), magrittr (>= 1.5), purrr (>= 0.3.2), rlang (>= 0.3.4), stats, stringr (>= 1.4.0)

Suggests covr (>= 3.2.1), fgeo.analyze (>= 1.1.10), fgeo.x (>= 1.1.3), gridExtra (>= 2.3), knitr (>= 1.22), rmarkdown (>= 1.12), spelling (>= 2.1), testthat (>= 2.1.1)

Encoding UTF-8

Language en-US

RoxygenNote 7.2.1

NeedsCompilation no

Author Mauro Lepore [aut, ctr, cre] (<https://orcid.org/0000-0002-1986-7988>), CTFS-ForestGEO [cph, fnd]

Maintainer Mauro Lepore <maurolepore@gmail.com>

Repository CRAN

Date/Publication 2022-09-03 18:30:02 UTC

R topics documented:

`autoplot.fgeo_habitat` *Plot habitats.*

Description

Plot habitats.

Usage

```
## S3 method for class 'fgeo_habitat'
autoplot(object, ...)
```

Arguments

`object`	An object of class "fgeo_habitat" (see `fgeo_habitat` at https://forestgeo.github.io/fgeo/articles/siteonly/reference.html).
`...`	Not used (included for compatibility across methods).

Value

An object of class "ggplot".

See Also

Other plot functions: `autoplot.sp_elev()`, `autoplot_by_species.sp_elev()`, `elev()`, `plot_dbh_bubbles_by_quadrat`
`plot_tag_status_by_subquadrat()`, `sp_elev()`, `sp()`

Other autoplots: `autoplot.sp_elev()`, `elev()`, `sp_elev()`, `sp()`

Examples

```
assert_is_installed("fgeo.x")
habitats <- fgeo.x::habitat
autoplot(habitats)
```

`autoplot.sp_elev` *Plot species distribution and/or topography.*

Description

Plot the columns `sp` and/or `elev` of ForestGEO-like datasets of class 'sp' and/or 'sp_elev'.

- You can create a 'sp' `object` with:

  ```
  object <- sp(DATA-WITH-VARIABLE-sp)
  ```

- You can create an 'elev' `object` with:

  ```
  object <- elev(DATA-WITH-VARIABLE-elev)
  ```

- You can create a 'sp_elev' `object` with:

  ```
  object <- sp_elev(DATA-WITH-VARIABLE-sp, DATA-WITH-VARIABLE-elev)
  ```

See **Examples** below.

Usage

```
## S3 method for class 'sp_elev'
autoplot(
  object,
  fill = "sp",
  hide_fill_legend = FALSE,
  shape = 21,
  point_size = 3,
  facet = TRUE,
  contour_size = 0.5,
  low = "blue",
  high = "red",
  hide_color_legend = FALSE,
  bins = NULL,
  add_elevation_labels = TRUE,
  label_size = 3,
  label_color = "grey",
  xyjust = 1,
  fontface = "italic",
  xlim = NULL,
  ylim = NULL,
  custom_theme = NULL,
  ...
)
```

```
## S3 method for class 'sp'
autoplot(
  object,
  fill = "sp",
  hide_fill_legend = FALSE,
  shape = 21,
  point_size = 3,
  facet = TRUE,
  xlim = NULL,
  ylim = NULL,
  custom_theme = NULL,
  ...
)

## S3 method for class 'elev'
autoplot(
  object,
  contour_size = 0.5,
  low = "blue",
  high = "red",
  hide_color_legend = FALSE,
  bins = NULL,
  add_elevation_labels = TRUE,
  label_size = 3,
  label_color = "grey",
  xyjust = 1,
  fontface = "italic",
  xlim = NULL,
  ylim = NULL,
  custom_theme = NULL,
  ...
)
```

Arguments

object	An object created with sp(), elev(), or sp_elev().
fill	Character; either a color or "sp", which maps each species to a different color.
hide_fill_legend	Logical; TRUE hides the fill legend.
shape	A number giving point shape (as in graphics::points()). Passed to ggplot2::geom_point().
point_size	A number giving point size. Passed to ggplot2::geom_point().
facet	Logical; TRUE wraps multiple panels within the area of a single graphic plot.
contour_size	A number giving the size of the contour of elevation lines. Passed to ggplot2::stat_contour() (see ggplot2::geom_contour()).
low, high	A string giving a color of the elevation lines representing low and high elevation.
hide_color_legend	Logical; TRUE hides the color legend.

`bins`	A number giving the number of elevation lines to plot.
`add_elevation_labels`	Logical; `FALSE` hides elevation labels.
`label_size, label_color, fontface`	A number (`label_size`) or character string (`label_color` and `fontface`) giving the size, colour and fontface of the text labels for the elevation lines.
`xyjust`	A number to adjust the position of the text labels of the elevation lines.
`xlim, ylim`	A vector of length 2, for example `c(0, 500)`, giving the minimum and maximum limits of the vertical and horizontal coordinates.
`custom_theme`	A valid `ggplot2::theme()`. `NULL` uses the default theme `theme_default()`.
`...`	Not used (included for compatibility across methods).

Details

`autoplot(sp_elev(DATA-WITH-VARIABLE-sp)` (without elevation data) is equivalent to `autoplot(sp(DATA-WITH-VARIABL`

fgeo.plot wraps some functions from the **ggplot2** package. For more control you can use **ggplot2** directly.

Value

A "ggplot".

See Also

Other plot functions: `autoplot.fgeo_habitat()`, `autoplot_by_species.sp_elev()`, `elev()`, `plot_dbh_bubbles_by_quadrat()`, `plot_tag_status_by_subquadrat()`, `sp_elev()`, `sp()`

Other autoplots: `autoplot.fgeo_habitat()`, `elev()`, `sp_elev()`, `sp()`

Other functions to plot elevation: `autoplot_by_species.sp_elev()`, `elev()`, `sp_elev()`

Other functions to plot species: `autoplot_by_species.sp_elev()`, `sp_elev()`, `sp()`

Examples

```
assert_is_installed("fgeo.x")

# Species -----------------------------------------------------------------

# Small dataset with a few species for quick examples
census <- fgeo.x::tree5 %>%
  subset(sp %in% c("PREMON", "CASARB"))

autoplot(sp(census))

# Skip R CMD check for speed

# Customize
autoplot(sp(census), point_size = 1)

# Elevation ---------------------------------------------------------------
```

```
elevation <- fgeo.x::elevation
autoplot(elev(elevation))

# Skip R CMD check for speed
# Same as `autoplot(elev(elevation))`
autoplot(elev(elevation$col))

# Customize
autoplot(elev(elevation), contour_size = 1)

# Species and elevation ------------------------------------------------
autoplot(sp_elev(census, elevation), facet = FALSE, point_size = 1)
```

`autoplot_by_species.sp_elev`

List plots of species distribution and topography (good for pdf output).

Description

These functions extend `autoplot.sp()` and `autoplot.elev()` and return not a single plot but a list of plots. They are particularly useful if you want to print a *.pdf* file with one plot per page. They automatically plot the variables `sp` and `elev` of a ForestGEO-like dataset of class 'sp' or 'sp_elev'.

- Create a 'sp' `object` with:

  ```
  object <- sp(DATA-WITH-VARIABLE-sp)
  ```

- Create a 'sp_elev' `object` with:

  ```
  object <- sp_elev(DATA-WITH-VARIABLE-sp, DATA-WITH-VARIABLE-elev)
  ```

See sections **Usage** and **Examples**.

Usage

```
## S3 method for class 'sp_elev'
autoplot_by_species(
  object,
  species = "all",
  fill = "black",
  shape = 21,
  point_size = 3,
  contour_size = 0.5,
  low = "blue",
  high = "red",
  hide_color_legend = FALSE,
```

```
  bins = NULL,
  add_elevation_labels = TRUE,
  label_size = 3,
  label_color = "grey",
  xyjust = 1,
  fontface = "italic",
  xlim = NULL,
  ylim = NULL,
  custom_theme = NULL,
  ...
)

## S3 method for class 'sp'
autoplot_by_species(
  object,
  species = "all",
  fill = "black",
  shape = 21,
  point_size = 3,
  hide_color_legend = FALSE,
  xlim = NULL,
  ylim = NULL,
  custom_theme = NULL,
  ...
)
```

Arguments

`object`	An object created with `sp()` or `sp_elev()`.
`species`	A character vector giving values in the column `sp`. The output will be a list with as many plots as elements in this vector. The string "all" (default) plots all unique values of `sp`.
`fill`	Character; either a color or "sp", which maps each species to a different color.
`shape`	A number giving point shape (as in `graphics::points()`). Passed to `ggplot2::geom_point()`.
`point_size`	A number giving point size. Passed to `ggplot2::geom_point()`.
`contour_size`	A number giving the size of the contour of elevation lines. Passed to `ggplot2::stat_contour()` (see `ggplot2::geom_contour()`).
`low, high`	A string giving a color of the elevation lines representing low and high elevation.
`hide_color_legend`	Logical; `TRUE` hides the color legend.
`bins`	A number giving the number of elevation lines to plot.
`add_elevation_labels`	Logical; `FALSE` hides elevation labels.
`label_size, label_color, fontface`	A number (`label_size`) or character string (`label_color` and `fontface`) giving the size, colour and fontface of the text labels for the elevation lines.

xyjust	A number to adjust the position of the text labels of the elevation lines.
xlim, ylim	A vector of length 2, for example c(0, 500), giving the minimum and maximum limits of the vertical and horizontal coordinates.
custom_theme	A valid ggplot2::theme(). NULL uses the default theme theme_default().
...	Not used (included for compatibility across methods).

Details

autoplot_by_species(sp_elev(DATA-WITH-VARIABLE-sp) (without elevation data) is equivalent to autoplot_by_species(sp(DATA-WITH-VARIABLE-sp)).

fgeo.plot wraps some functions from the **ggplot2** package. For more control you can use **ggplot2** directly.

Value

A list of objects of class "ggplot".

See Also

autoplot(), sp(), sp_elev().

Other plot functions: autoplot.fgeo_habitat(), autoplot.sp_elev(), elev(), plot_dbh_bubbles_by_quadrat(), plot_tag_status_by_subquadrat(), sp_elev(), sp()

Other functions to plot elevation: autoplot.sp_elev(), elev(), sp_elev()

Other functions to plot species: autoplot.sp_elev(), sp_elev(), sp()

Examples

```
assert_is_installed("fgeo.x")

# Species -----------------------------------------------------------------
# Small dataset with a few species for quick examples
census <- fgeo.x::tree6_3species

# Showing only two species for speed
autoplot_by_species(sp(census))[1:2]

# To print all plots in a .pdf see `?pdf()`
autoplot_by_species(sp(census))

# Species and elevation (optional) ----------------------------------------

# Species and elevation
elevation <- fgeo.x::elevation
autoplot_by_species(sp_elev(census, elevation))
```

elev *Allow autoplotting the column* elev.

Description

Allow autoplotting the column elev.

Usage

```
elev(elev)
```

Arguments

elev	A ForestGEO-like elevation list or its col dataframe (with the column elev).

Value

An S3 object of class 'elev'.

See Also

autoplot.elev().

Other plot functions: autoplot.fgeo_habitat(), autoplot.sp_elev(), autoplot_by_species.sp_elev(), plot_dbh_bubbles_by_quadrat(), plot_tag_status_by_subquadrat(), sp_elev(), sp()

Other autoplots: autoplot.fgeo_habitat(), autoplot.sp_elev(), sp_elev(), sp()

Other functions to construct fgeo classes: sp_elev(), sp()

Other functions to plot elevation: autoplot.sp_elev(), autoplot_by_species.sp_elev(), sp_elev()

Examples

```
assert_is_installed("fgeo.x")

inherits(elev(fgeo.x::elevation), "elev")
inherits(elev(fgeo.x::elevation$col), "elev")
```

plot_dbh_bubbles_by_quadrat *List dbh bubble-plots by quadrat (good for .pdf output).*

Description

List dbh bubble-plots by quadrat (good for .pdf output).

Usage

```
plot_dbh_bubbles_by_quadrat(
  vft,
  title_quad = "Site Name, YYYY, Quadrat:",
  header = header_dbh_bubbles(),
  theme = theme_dbh_bubbles(),
  lim_min = 0,
  lim_max = 20,
  subquadrat_side = 5,
  tag_size = 2,
  move_edge = 0,
  status_d = "dead"
)
```

Arguments

vft	A ForestGEO ViewFullTable (dataframe).
title_quad	A string to use as a title.
header	A string to use as a header (see headers).
theme	An object of class "theme".
lim_min, lim_max	Minimum and maximum limits of the plot area.
subquadrat_side	Length in meters of the side of a subquadrat.
tag_size	A number giving tag size. Passed to ggrepel::geom_text_repel.
move_edge	A number to adjust the extension of the grid lines beyond the plot limits.
status_d	A character string indicating the value of the variable status that corresponds to dead stems.

Value

A list which each element is a plot of class ggplot.

See Also

Other plot functions: autoplot.fgeo_habitat(), autoplot.sp_elev(), autoplot_by_species.sp_elev(), elev(), plot_tag_status_by_subquadrat(), sp_elev(), sp()

Other functions to list plots from ForestGEO ViewFullTable: plot_tag_status_by_subquadrat()

Other functions to plot dbh bubbles: header_dbh_bubbles(), theme_dbh_bubbles()

Examples

```
assert_is_installed("fgeo.x")

# Create a small VieFullTable
first_n <- function(x, n) x %in% sort(unique(x))[1:n]
```

```
small_vft <- fgeo.x::vft_4quad %>%
  dplyr::filter(first_n(CensusID, 1) & first_n(QuadratID, 2)) %>%
  dplyr::sample_n(50)

plot_dbh_bubbles_by_quadrat(small_vft)

# To print all plots into a .pdf file see `?pdf()`
plot_dbh_bubbles_by_quadrat(small_vft)

# Be careful if subsetting by DBH: You may unintentionally remove dead trees.
# You should expliciety inlcude missing `DBH` values with `is.na(DBH)`
include_missing_dbh <- subset(small_vft, DBH > 20 | is.na(DBH))
plot_dbh_bubbles_by_quadrat(include_missing_dbh)

# Customizing the maps ------------------------------------------------
# A custom title and header
myheader <- paste(
  " ",
  "Head column 1                    Head column 2                        ",
  " ",
  " .....................................................................",
  " .....................................................................",
  sep = "\n"
)

plot_dbh_bubbles_by_quadrat(
  small_vft,
  title_quad = "My Site, 2018. Quad:",
  header = myheader
)

# Skip R CMD check for speed

# Tweak the theme with ggplot
library(ggplot2)

plot_dbh_bubbles_by_quadrat(
  small_vft,
  title_quad = "My Site, 2018. Quad:",
  header = header_dbh_bubbles("spanish"),
  tag_size = 3,
  theme = theme_dbh_bubbles(
    axis.text = NULL, # NULL shows axis.text; element_blank() doesn't.
    plot.title = element_text(size = 15),
    plot.subtitle = element_text(size = 5),
    panel.background = element_rect(fill = "grey")
  )
)
```

`plot_tag_status_by_subquadrat` *List plots of tree-tag status by subquadrat (good for .pdf output).*

Description

This function plots tree tags by status and outputs a list of plots that can be printed on a .pdf file. Each plot shows four subquadrats within a quadrat. The symbols on the plot represent the status of each tree – not the status of each stem. Although you should likely provide data of only one or two censuses, `plot_tag_status_by_subquadrat()` will summarize the data to reduce overplotting. The data on the plot summarizes the history of each stem across all censuses provided. Each tag will appear in the plot only once or twice:

- A tag will appear once if it belongs to a tree which status was unique across all censuses provided – either "alive" or "dead".
- A tag will appear twice if it belongs to a tree which status was "alive" in at least one census, and also "dead" in at least one other census. This feature avoids unintentional overplotting and makes interpreting the plot easier.

Usage

```
plot_tag_status_by_subquadrat(
  vft,
  x_q = 20,
  x_sq = 5,
  y_q = 20,
  y_sq = 5,
  subquad_offset = NULL,
  bl = 1,
  br = 2,
  tr = 3,
  tl = 4,
  title_quad = "Site Name, YYYY. Quadrat:",
  show_page = TRUE,
  show_subquad = TRUE,
  point_shape = c(19, 4),
  point_size = 1.5,
  tag_size = 3,
  header = header_tag_status(),
  theme = theme_tag_status(),
  move_edge = 0
)
```

Arguments

`vft`	A ForestGEO ViewFullTable (dataframe).
`x_q, y_q`	Size in meters of a quadrat's side. For ForestGEO sites, a common value is 20.

`x_sq, y_sq` Size in meters of a subquadrat's side. For ForestGEO-CTFS sites, a common value is 5.

`subquad_offset` `NULL` or `-1`. `NULL` defines the first column of subquadrats as 1. `-1` defines the first column of subquadrats as 0.

```
subquad_offset = NULL     subquad_offset = -1
---------------------     -------------------
     14 24 34 44               04 14 24 34
     13 23 33 43               03 13 23 33
     12 22 32 42               02 12 22 32
     11 21 31 41               01 11 21 31
```

`bl, br, tr, tl` Number or character giving the label of the four subquadrats on each or the four divisions of a quadrat: bottom left (bl), bottom right (br), top right (tr), and top left (tl).

`title_quad` A string to use as a title.

`show_page` Logical; `FALSE` removes the page label from the plot title.

`show_subquad` Logical; `FALSE` removes subquadrat labels on each plot.

`point_shape` A vector of two numbers giving the shape of the points to plot (see possible shapes in the documentation of ?`graphics::points()`, under the section entitled *'pch' values*).

`point_size` A number giving points size. Passed to `ggplot2::geom_point()`.

`tag_size` A number giving tag size. Passed to ggrepel::geom_text_repel.

`header` A string to use as a header (see headers).

`theme` An object of class "theme".

`move_edge` A number to adjust the extension of the grid lines beyond the plot limits.

Value

A list of objects of class "ggplot".

Acknowledgment

Useful ideas and guidance came from Suzanne Lao, Stuart Davis, Shameema Jafferjee Esufali, David Kenfack and Anudeep Singh. Anudeep Sinh also wrote the algorithm to calculate subquadrats.

See Also

`graphics::points()`, `ggplot2::geom_point()`, `ggplot2::theme()` `header_tag_status()`, `theme_tag_status()`, `fgeo.tool::add_subquad()`, ggrepel::geom_text_repel.

Other plot functions: `autoplot.fgeo_habitat()`, `autoplot.sp_elev()`, `autoplot_by_species.sp_elev()`, `elev()`, `plot_dbh_bubbles_by_quadrat()`, `sp_elev()`, `sp()`

Other functions to list plots from ForestGEO ViewFullTable: `plot_dbh_bubbles_by_quadrat()`

Other functions to plot tag status: `header_tag_status()`, `theme_tag_status()`

Examples

```
assert_is_installed("fgeo.x")

# Create a small VieFullTable
first <- function(x) x %in% sort(unique(x))[1]
small_vft <- subset(fgeo.x::vft_4quad, first(CensusID) & first(QuadratID))

p <- plot_tag_status_by_subquadrat(small_vft)
# Showing only two sub-quadtrats
p[1:2]

# To print all plots into a .pdf file see `?pdf()`
plot_tag_status_by_subquadrat(small_vft)

# Be careful if filtering by DBH: You may unintentionally remove dead trees.
# * If you filter by `DBH`, you loose the dead trees becaue their `DBH = NA`
# * You should explicietly inlcude missing DBH values with `is.na(DBH)`
include_missing_dbh <- subset(small_vft, DBH > 20 | is.na(DBH))
p <- plot_tag_status_by_subquadrat(include_missing_dbh)
# Showing only the first plot to keep the output short
p[[1]]

# Customizing the maps ------------------------------------------------
# Common tweaks
p <- plot_tag_status_by_subquadrat(
  small_vft,
  title_quad = "BCI 2012. Quadrat: ",
  bl = "bottom-left", br = "bottom-right", tr = "top-right", tl = "top-left",
  header = "Line 1: _________\nLine 2:\nLine 3:.....................",
  subquad_offset = -1,
  point_size = 3, point_shape = c(17, 6),
  tag_size = 2,
  move_edge = 0.5
)
p[[1]]

# Skip R CMD check for speed

p <- plot_tag_status_by_subquadrat(
  small_vft,
  show_page = FALSE,
  show_subquad = FALSE
)
p[[1]]

# Themes
library(ggplot2)

p <- plot_tag_status_by_subquadrat(small_vft, theme = theme_gray())
p[[1]]

# Tweaking the default theme of plot_tag_status_by_subquadrat()
```

```
# For many more options see ?ggplot2::theme
small_tweak <- theme_tag_status(legend.position = "bottom")
p <- plot_tag_status_by_subquadrat(small_vft, theme = small_tweak)
p[[1]]
```

---

sp *Allow autoplotting the column* sp.

Description

Allow autoplotting the column sp.

Usage

```
sp(sp)
```

Arguments

sp	A ForestGEO-like dataframe with the column sp.

Value

An S3 object of class 'sp'.

See Also

autoplot.sp().

Other plot functions: autoplot.fgeo_habitat(), autoplot.sp_elev(), autoplot_by_species.sp_elev(), elev(), plot_dbh_bubbles_by_quadrat(), plot_tag_status_by_subquadrat(), sp_elev()

Other autoplots: autoplot.fgeo_habitat(), autoplot.sp_elev(), elev(), sp_elev()

Other functions to construct fgeo classes: elev(), sp_elev()

Other functions to plot species: autoplot.sp_elev(), autoplot_by_species.sp_elev(), sp_elev()

Examples

```
assert_is_installed("fgeo.x")

inherits(sp(fgeo.x::stem5), "sp")
```

sp_elev *Allow autoplotting the columns* sp *and* elev.

Description

Allow autoplotting the columns sp and elev.

Usage

```
sp_elev(sp, elev = NULL)
```

Arguments

sp	A ForestGEO-like dataframe with the column sp.
elev	A ForestGEO-like elevation list or its col dataframe – with the column elev. The datasets you pass to sp and elev should come from the same forest plot. This is not compulsory but not doing so is most likely a mistake.

Value

An S3 object of class 'sp_elev'.

See Also

autoplot.sp_elev().

Other plot functions: autoplot.fgeo_habitat(), autoplot.sp_elev(), autoplot_by_species.sp_elev(), elev(), plot_dbh_bubbles_by_quadrat(), plot_tag_status_by_subquadrat(), sp()

Other autoplots: autoplot.fgeo_habitat(), autoplot.sp_elev(), elev(), sp()

Other functions to construct fgeo classes: elev(), sp()

Other functions to plot elevation: autoplot.sp_elev(), autoplot_by_species.sp_elev(), elev()

Other functions to plot species: autoplot.sp_elev(), autoplot_by_species.sp_elev(), sp()

Examples

```
assert_is_installed("fgeo.x")

species_from_luquillo <- fgeo.x::stem5
elevation_from_luquillo <- fgeo.x::elevation

species_elevation <- sp_elev(species_from_luquillo, elevation_from_luquillo)
inherits(species_elevation, "sp_elev")
```

Index