Package ‘fireexposuR’

May 27, 2025

Type Package

Title Compute and Visualize Wildfire Exposure

Version 1.1.0

Description Methods for computing and visualizing wildfire ignition exposure and directional vulnerability that are published in a series of scientific publications are automated by the functions in this package. See Beverly et al. (2010) <doi:10.1071/WF09071>, Beverly et al. (2021) <doi:10.1007/s10980-020-01173-8>, and Beverly and Forbes (2023) <doi:10.1007/s11069-023-05885-3> for background and methodology.

License GPL (>= 3)

Imports dplyr, geosphere, ggplot2, ggspatial, magrittr, maptiles, MultiscaleDTM, rlang, terra, tidyr, tidyselect, tidyterra

Encoding UTF-8

RoxygenNote 7.3.2

URL `https://github.com/ropensci/fireexposuR`, `https://docs.ropensci.org/fireexposuR/`

BugReports `https://github.com/ropensci/fireexposuR/issues`

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

VignetteBuilder knitr

NeedsCompilation no

Author Air Forbes [aut, cre] (ORCID: <https://orcid.org/0000-0002-9842-7648>), Jennifer Beverly [ths]

Maintainer Air Forbes <heyairf@gmail.com>

Repository CRAN

Date/Publication 2025-05-27 08:00:02 UTC

Contents

`fire_exp` *Compute the wildfire exposure metric*

Description

`fire_exp()` returns a SpatRaster of wildfire exposure values calculated using the input hazard fuel raster for a specified transmission distance.

Usage

```
fire_exp(hazard, tdist = c("l", "s", "r"), no_burn)
```

Arguments

`hazard`	a SpatRaster that represents hazardous fuels for the transmission distance specified in tdist
`tdist`	a character vector, can be: `"l"` for long-range embers (Default), `"s"` for short-range embers or, `"r"` for radiant heat
`no_burn`	(optional) a SpatRaster that represents the non-burnable landscape. Any cells that cannot receive wildfire (e.g. open water, rock) and any cells that are not natural (e.g. built environment, irrigated agricultural areas) should be of value 1, all other cells should be NODATA. This parameter should be provided if preparing data for `fire_exp_validate()`

Details

This function is the primary function in this package, the output from this function serves as the first input to all other functions in this package.

Background:

Wildfire exposure was first proposed for community scale assessments by Beverly et al. (2010). Wildfire entry points to the built environment were identified by evaluating the surrounding wildland fuels as potential ignition sources. These methods were adapted and validated by Beverly et al. (2021) for landscape scale assessments. Visual aids to better understand these concepts are available in the Wildfire Exposure Assessment Guidebook (FireSmart Canada 2018) which includes intuitive illustrations.

Three transmission distances were defined in Beverly et al. (2010) for different scales of wildfire ignition processes. Different fuel types are capable of transmitting fire at different scales. The transmission distances define a potential maximum spread distance from an ignition source. If the default distances do not accurately represent fire behaviour in your area of interest you can adjust them with `fire_exp_adjust()`.

Radiant heat:

The default transmission distance for radiant heat is 30 meters. At this scale of wildfire transmission wildfire can spread from direct flame contact or thermal radiation warming of fuels.

Ember spotting:

Embers (AKA firebrands) can be carried by wind beyond the active fire front. To determin two different ember spreading distances Beverly et al. (2010) reviewed fire-spot distance observations and applied predictive models to verify these distances were a reasonable assumption.

Short-range embers:

The default transmission distance for short-range embers is 100 meters. Some fuel types have the potential to produce embers over short distances. An example fuel type that has the potential to transmit short-range embers is a deciduous forest. Fuels considered hazardous to short-range ember ignition will also be hazardous to transmit fire via radiant heat or direct flame contact.

Long-range embers:

The default transmission distance for long-range embers is 500 meters. Some fuel types have the potential to transmit embers across large distances. This landscape scale process is often a contributor to large, fast-moving fires. Any fuels that are considered hazardous for long-range ember spotting could also transmit fire by short-range ember spotting and radiant heat. An example fuel type that has the potential to transmit long-range embers is a mature pine stand. Although it is possible for embers from certain fuel types to be carried much farther than the defined 500 meter transmission distance under extreme conditions, these cases are relatively uncommon.

The 500 meter spread distance has been further validated across Alberta (Beverly et al. 2021), in Alaska (Schmidt et al. 2024), across the entire Canadian landbase (manuscript in preparation), and in Portugal (manuscript in preparation).

Technical:

How the metric is calculated:

The wildfire exposure metric is calculated for each cell individually using a focal window of the surrounding cells within the specified transmission distance. The proportion of cells within the assessment window that are classed as hazardous is returned. The wildfire exposure metric has a range of 0-1. A wildfire exposure value of 0.5 can be interpreted as 50% of cells within the specified transmission distance area have the potential to spread fire to the assessment cell. A value of 0.5 can also be interpreted in the other direction, a fire in the assessment cell could potentially spread to 50% of the surrounding cells within the specified transmission distance.

Input features:
An input hazard raster must be prepared by the user in accordance with the intended use. First, refer to the Get Started vignette by running `vignette("fireexposuR")` to determine what the data requirements are for the intended application. Then refer to the Preparing Input Data vignette by running `vignette("prep-input-data")` for lots of recommendations, advice, and examples.
A separate hazard raster should be prepared for each of the transmission distances of interest. There are also minimum spatial resolution and extent requirements for each transmission distances.
Long-range embers:

- minimum raster resolution is 150 meters
- The dimensions of the data must be wider/taller than 1000 meters because 500 meters of data will be lost along the perimeter due to edge effects

Short-range embers:

- minimum raster resolution is 33 meters
- The dimensions of the data must be wider/taller than 200 meters because 100 meters of data will be lost along the perimeter due to edge effects

Radiant heat:

- minimum raster resolution is 10 meters
- The dimensions of the data must be wider/taller than 60 meters because 30 meters of data will be lost along the perimeter due to edge effects

Spatial Reference:
The exposure raster will be returned in the same CRS as the input hazard layer. A crs must be defined if the outputs will be used in other functions in this package.

Value

A SpatRaster object of exposure values between 0-1

References

Beverly JL, McLoughlin N, Chapman E (2021) A simple metric of landscape fire exposure. *Landscape Ecology* **36**, 785-801. doi:10.1007/s10980020011738

Beverly JL, Bothwell P, Conner JCR, Herd EPK (2010) Assessing the exposure of the built environment to potential ignition sources generated from vegetative fuel. *International Journal of Wildland Fire* **19**, 299-313. doi:10.1071/WF09071

FireSmart Canada (2018) Wildfire exposure assessment guidebook. Available here

Hijmans R (2024). *terra: Spatial Data Analysis*. R package version 1.7-78, CRAN.

Schmidt JI, Ziel RH, Calef MP, Varvak A (2024) Spatial distribution of wildfire threat in the far north: exposure assessment in boreal communities. *Natural Hazards* **120**, 4901-4924. doi:10.1007/s11069023063654

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# compute long range exposure
fire_exp(hazard, tdist = "l")
```

`fire_exp_adjust` *Compute the wildfire exposure metric with custom transmission distances*

Description

For advanced users. Adjust the transmission distances from the defaults used in `fire_exp()`.

Usage

```
fire_exp_adjust(hazard, tdist, no_burn)
```

Arguments

`hazard`	a SpatRaster that represents hazardous fuels for the transmission distance specified in `tdist`
`tdist`	Numeric, transmission distance in meters
`no_burn`	(Optional) SpatRaster that represents the non-burnable landscape. Any cells that cannot receive wildfire (e.g. open water, rock) and any cells that are not natural (e.g. built environment, irrigated agricultural areas) should be of value 1, all other cells should be NODATA. This parameter should be provided if preparing data for `fire_exp_validate()`

Details

If the transmission distances from the wildfire exposure literature are not representative of the wildland fuels in your area of interest, this function can be used to change the transmission distance to a custom distance. It is highly recommended that any exposure layers produced with this function are validated with observed fire history using the `fire_exp_validate()` function.

Spatial Reference:

The exposure raster will be returned in the same CRS as the input hazard layer. A crs must be defined if the outputs will be used in other functions in this package.

Spatial resolution:

For a specified transmission distance, the spatial resolution must be at least 3 times finer. For example, for a transmission distance of 300 meters the input data should have a resolution of 100 meters or finer.

Value

SpatRaster object of exposure values

See Also

fire_exp() for background information

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# compute long range exposure with custom disance of 800 m
exposure800 <- fire_exp_adjust(hazard, tdist = 800)
```

---

fire_exp_dir *Conduct a directional exposure assessment*

Description

fire_exp_dir() returns a SpatVector of linear transects toward a value. Transects are assessed as viable wildfire pathways by intersecting them with areas of high exposure.

Usage

```
fire_exp_dir(
  exposure,
  value,
  t_lengths = c(5000, 5000, 5000),
  interval = 1,
  thresh_exp = 0.6,
  thresh_viable = 0.8,
  table = FALSE
)
```

Arguments

exposure	SpatRaster (e.g. from fire_exp())
value	Spatvector of value as a point or simplified polygon
t_lengths	(Optional) A vector of three numeric values. The length of each transect starting from the value in meters. The default is c(5000, 5000, 5000).
interval	(Optional) Numeric. The degree interval at which to draw the transects for analysis. Can be one of 0.25, 0.5, 1, 2, 3, 4, 5, 6, 8, or 10 (factors of 360, ensures even spacing). The default is 1.

`thresh_exp`	(optional) Numeric. The exposure value to use to define high exposure as a proportion. Must be between 0-1. The default is `0.6`.
`thresh_viable`	(optional) Numeric. The minimum intersection of a transect with a high exposure patch to be defined as a viable pathway as a proportion. Must be between 0-1. The default is `0.8`.
`table`	Boolean, when `TRUE`: returns a table instead of a feature. The default is `FALSE`.

Details

`fire_exp_dir()` automates the directional vulnerability assessment methods documented in Beverly and Forbes (2023). This analysis is used to assess linear wildfire vulnerability on the landscape in a systematic radial sampling pattern. This is a landscape scale process, so the exposure raster used should also be landscape scale. Use `tdist = "l"` when preparing data with `fire_exp()` for use with this function. See `fire_exp()` details for more information.

The output line features will have the attribute 'viable' which can be used to visualize the pathways. Outputs can be visualized with `fire_exp_dir_plot()`, `fire_exp_dir_map()`, or exported as a spatial feature.

Spatial Reference:

The inputs for the exposure and value layer must have the same coordinate reference system (CRS) defined. The transects will be returned in the same CRS as the inputs.

This function draws the transects by calculating the end point of a transect by finding the shortest path along the WGS84 (EPSG:4326) ellipsoid at a given bearing and distance. The `value` input is reprojected from the input CRS to latitude and longitude using WGS84 for the calculations. After the transects are created, they are projected to match the CRS of the input exposure and value layer. The lengths of the projected transects will be effected by the scale factor of the input CRS; however, the geodesic lengths are maintained.

Value feature:

The value feature can be provided as a point or a simplified polygon.

If using a point feature the analysis can be sensitive to the placement of a point. For example, if using a point to represent a large town a point placed at the centroid will likely have different outputs than a point placed at the edge of the community due to the arrangement of lower exposure values typical of a built environment.

An option to use a simplified polygon has been added to this function for values that may be too large to represent with a single point. The polygon should be drawn with the consideration of the following or the function will not be able to run. The polygon must be a single part polygon with no holes. The polygon should have a smooth and simple shape, ideally circular or ellipsoidal. The polygon should also have an approximate radius of less than 5000 meters from the center. If the area of interest is larger than this then consider using multiple assessments.

Default Values:

The default values are based on the methods used and validated in Alberta, Canada by Beverly and Forbes (2023). Options have been added to the function to allow these values to be adjusted by the user if required. Adjusting these values can lead to unexpected and uncertain results.

Adjusting the transects:

The drawing of the transects can be customized by varying the intervals and segment lengths if desired. Adjustments to the `interval` and `t_lengths` parameters will effect how much of the exposure data is being captured by the transects. If both parameters are being adjusted some trigonometry might be required to find the optimal combination of the parameters to ensure distances between the transects make sense for the intended application. The resolution of the exposure raster may also be a factor in these decisions.

Interval:
The interval parameter defines how many transects are drawn. The default of `1` draws a transect at every whole degree from 1-360. This outputs a total of 1080 transects, 360 for each segment. Increasing the interval will output less transects and process faster. When the interval is increased, the distance between the ends of the transects will also be increased. For example: the terminus of 15000 meter transects (the default) drawn every 1 degree (the default) will be approximately 250 meters apart, but if drawn at 10 degree intervals will be approximately 2500 meters apart. Larger intervals will increase speed and processing time, but might not capture potential pathways between transects farther from the value.

Lengths:
The t_lengths parameter allows a custom distance to be defined for the three transect segments in meters. Lengths can be increased or decreased. The segments can also be different lengths if desired.

Adjusting the thresholds:

Threshold adjustments should only be considered if validation within the area of interest have been done. The function `fire_exp_validate()` has been included with this package to make the process easier, but still requires significant time, data, and understanding.

High exposure:
The `thresh_exp` parameter can be adjusted to define the minimum exposure value to be considered for the directional assessment. The default value of `0.6` is based on the findings of Beverly et al. (2021) who showed that observed fires burned preferentially in areas where wildfire exposure values exceed 60%. Adjusting this value is only recommended after a validation of wildfire exposure has been conducted in the area of interest.

Viability:
The `thresh_viable` parameter defines the minimum intersection with high exposure areas to be considered a viable pathway. The default value of `0.8` was determined by Beverly and Forbes (2023) by drawing continuous linear transects within burned areas to represent observed pathways. It was found that the average intersection with patches of pre-fire high exposure was 80%. This methodology could be repeated in the users area of interest.

Value

a SpatVector of the transects with the attributes: 'deg' = degree, 'seg' = segment, and 'viable'. The transects will be returned with the same CRS as the input features.

If `table = TRUE`: a data frame is returned instead with an additional attribute 'wkt', which is a Well Known Text (WKT) string of transect geometries (coordinate reference system: WGS84).

References

Beverly JL, Forbes AM (2023) Assessing directional vulnerability to wildfire. *Natural Hazards* **117**, 831-849. doi:10.1007/s11069023058853

Beverly JL, McLoughlin N, Chapman E (2021) A simple metric of landscape fire exposure. *Landscape Ecology* **36**, 785-801. doi:10.1007/s10980020011738

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# generate an example point
point_wkt <- "POINT (400000 6050000)"
point <- terra::vect(point_wkt, crs = hazard)

# compute exposure metric
exposure <- fire_exp(hazard)

# assess directional exposure
fire_exp_dir(exposure, point)
```

---

fire_exp_dir_map *Map directional exposure*

Description

`fire_exp_dir_map()` plots directional exposure transects onto a map.

Usage

```
fire_exp_dir_map(
  transects,
  value,
  zoom_level = 10,
  labels,
  title = "Directional Vulnerability"
)
```

Arguments

`transects`	SpatVector. Output from `fire_exp_dir()`
`value`	(Optional) SpatVector. Adds the value to the map. Use the same value feature used to generate the transects with `fire_exp_dir()`
`zoom_level`	(Optional). Numeric. set the zoom level for the base map tile. See details. The default is `10`.
`labels`	(Optional) a vector of three strings. Custom formatting for the distances in the legend. If left blank, the function will automatically label the distances in meters.
`title`	(Optional) String. A custom title for the plot. The default is `"Directional Vulnerability"`

Details

This function returns a standardized map with basic cartographic elements.

The plot is returned as a ggplot object which can be exported/saved to multiple image file formats.

Spatial reference:
This function dynamically pulls map tiles for a base map. The inputs are projected to WGS 84/Pseudo-Mercator (EPSG:3857) to align them with the map tiles.

Zoom level:
The map tile zoom level may need to be adjusted. If the base map is blurry, increase the zoom level. Higher zoom levels will slow down the function, so only increase if necessary. Reference the OpenStreetMap Wiki for more information on zoom levels.

Value

a map of directional exposure transects as a ggplot object

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# generate an example point
point_wkt <- "POINT (400000 6050000)"
point <- terra::vect(point_wkt, crs = hazard)

# compute exposure metric
exposure <- fire_exp(hazard)

# generate transects
transects <- fire_exp_dir(exposure, point)

# map with customized distance labels
fire_exp_dir_map(transects, labels = c("5 km", "10 km", "15 km"),
                 zoom_level = 9)
```

`fire_exp_dir_multi` *Summarize or plot directional load for multiple values*

Description

`fire_exp_dir_multi()` summarizes the directional vulnerability load for multiple points in a study area in a table or a plot.

Usage

```
fire_exp_dir_multi(exposure, values, plot = FALSE, full = FALSE, title, ...)
```

Arguments

`exposure`	SpatRaster from `fire_exp()`
`values`	Spatvector of value as a point or simplified polygon
`plot`	Boolean, when `TRUE`: returns a standardized directional plot. The default is `FALSE`.
`full`	Boolean. Ignored when `plot = FALSE`. When `TRUE`: all 3 transect segments must be viable. when `FALSE`: only the segments from seg2 and seg3 are considered (Default)
`title`	(Optional) String. Ignored when `plot = FALSE`. A custom title for the plot. The default is `"Directional Vulnerability for Multiple Values"`
`...`	arguments passed to `fire_exp_dir()`.

Details

This function summarizes multiple directional vulnerability assessments into a single table or plot. The plot is based on the methods presented in Beverly and Forbes 2023. For each degree, the frequency of input values with a continuous pathway at that trajectory is found. This summary can be useful in identifying trends in directional exposure to values within a regional area of interest.

Continuous pathways can be assessed for the full span of all three directional assessment transect segments, or limited to the outer two segments with the `full` parameter. If the values being assessed are variable sizes and being represented as points, it is recommended this parameter remains set to `FALSE`. The inner segment is sensitive to the size of the value when a point is used. Adjusting the parameters for `fire_exp_dir()` is also supported. See details in `fire_exp_dir()` for more information.

Value

a data.frame of the features with attributes: value featureID, degree, seg1 (binary), seg2 (binary), seg3 (binary), full (binary), outer (binary). Unless:

- `plot = TRUE`: a standardized plot as a ggplot object

References

Beverly JL, Forbes AM (2023) Assessing directional vulnerability to wildfire. *Natural Hazards* **117**, 831-849. doi:10.1007/s11069023058853

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# generate 10 random example points within the hazard extent
e <- terra::buffer(terra::vect(terra::ext(hazard), crs = hazard), -15500)
points <- terra::spatSample(e, 10)

# compute exposure metric
```

```
exposure <- fire_exp(hazard)

# plot directional load for multiple points
fire_exp_dir_multi(exposure, points, plot = TRUE, interval = 10)
```

`fire_exp_dir_plot` *Plot directional exposure in a radial plot*

Description

`fire_exp_dir_plot()` plots the viable directional exposure pathways identified with `fire_exp_dir()` in a standardized radial plot.

Usage

```
fire_exp_dir_plot(transects, labels, title = "Directional Vulnerability")
```

Arguments

`transects`	SpatVector (output from `fire_exp_dir()`)
`labels`	(Optional) a vector of three strings. Custom formatting for the distance labels on the transect segments. If left blank, the function will automatically label the distances in meters.
`title`	(Optional) String. A custom title for the plot. The default is `"Directional Vulnerability"`

Details

The radial plot produced by this function is based on the figures presented in Beverly and Forbes (2023). The plots put the transect origin (the value) at the center as a point, and labels the distances from the value at the end of the transect segments. If the value used to generate the transects was a polygon feature, the transect origins will still be drawn as a center point.

The plot is returned as a ggplot object which can be exported/saved to multiple image file formats.

Value

a ggplot object.

References

Beverly JL, Forbes AM (2023) Assessing directional vulnerability to wildfire. *Natural Hazards* **117**, 831-849. doi:10.1007/s11069023058853

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# generate an example point
point_wkt <- "POINT (400000 6050000)"
point <- terra::vect(point_wkt, crs = hazard)

# compute exposure metric
exposure <- fire_exp(hazard)

# generate transects
transects <- fire_exp_dir(exposure, point)


# radial plot
fire_exp_dir_plot(transects)

# customize labels
fire_exp_dir_plot(transects, labels = c("5 km", "10 km", "15 km"))
```

`fire_exp_extract` *Extract exposure values to features*

Description

`fire_exp_extract()` extracts the underlying exposure value for each feature in the values layer.

Usage

```
fire_exp_extract(exposure, values)
```

Arguments

`exposure`	SpatRaster (e.g. from `fire_exp()`)
`values`	Spatvector of points or polygons

Details

This function appends the underlying exposure value to the input feature as a new attribute. The values input can be provided as either points or polygons. The values should be singlepart features (i.e. the attribute table has one row per value). If the values are polygon features both the maximum and mean exposure is computed. Any values outside the extent of the exposure raster will be returned with an exposure of NA.

Outputs from this function can be visualized with `fire_exp_extract_vis()` or exported as a spatial feature.

Spatial Reference:

The inputs for the exposure and values layer must have the same coordinate reference system (CRS) defined. The transects will be returned in the same CRS as the inputs.

Scale:

The spatial resolution of the input exposure raster will effect the output. The exposure value returned by this function are based on the cell value underlying the feature in the values input. Note that if the resolution of the exposure raster is coarse, there may be multiple values within the same cell and the returned values will reflect this. For polygon features, the maximum and mean value of all cells within the boundary of the polygon are used. If the exposure raster is coarse, there may not be more than one cell within the polygon which will result in these values being the same.

Value

a SpatVector object with new attribute(s)

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# read example area of interest geometry
geom_file_path <- "extdata/polygon_geometry.csv"
geom <- read.csv(system.file(geom_file_path, package = "fireexposuR"))

# generate an area of interest polygon with the geometry
aoi <- terra::vect(as.matrix(geom), "polygons", crs = hazard)

# generate random points within the aoi polygon
points <- terra::spatSample(aoi, 100)

# compute exposure
exposure <- fire_exp(hazard)

fire_exp_extract(exposure, points)
```

`fire_exp_extract_vis` *Visualize exposure to values in a summary table or map*

Description

`fire_exp_extract_vis()` standardizes the visualization of outputs from `fire_exp_extract()` as a summary table or a map by classifying exposure into predetermined exposure classes.

Usage

```
fire_exp_extract_vis(
  values_ext,
  classify = c("local", "landscape", "custom"),
  class_breaks,
  method = c("max", "mean"),
  map = FALSE,
  zoom_level,
  title = "Classified Exposure to Values"
)
```

Arguments

`values_ext`	Spatvector of points or polygons from `fire_exp_extract()`
`classify`	character, either `"local"`, `"landscape"`, or `"custom"`, to specify classification scheme to use. The default is `"local"`. If set to `"custom"`: the parameter `class_breaks` must be used.
`class_breaks`	vector of numeric values between 0-1. Ignored unless `classify = "custom"`. See details.
`method`	character, either `"max"` or `"mean"`. If `values_ext` are polygons the default is `"max"`.This parameter is ignored when `values_ext` are point features.
`map`	Boolean. When `TRUE`, a map is returned as a ggplot object. The default is `FALSE`.
`zoom_level`	(Optional). Numeric. Ignored when `map = FALSE`. set the zoom level for the base map tile. See details. Defaults if: • `classify = "local"` or `"custom"` the zoom level default is `12` • `classify = "landscape"` the zoom level default is `7`
`title`	(Optional) String. Ignored when `map = FALSE`. A custom title for the plot. The default is `"Classified Exposure to Values"`

Details

This function visualizes the outputs from `fire_exp_extract()` with classes. Classes can be chosen from the pre-set `"local"` and `"landscape"` options, or customized. To use a custom classification scheme, it should be defined with a list of numeric vectors defining the upper limits of the breaks. A Nil class is added automatically for exposure values of exactly zero.

Local classification breaks are predefined as `c(0.15, 0.3, 0.45, 1)`:

- Nil (0)
- 0 - 0.15
- 0.15 - 0.3
- 0.3 - 0.45
- 0.45 - 1

#' Landscape classification breaks are predefined as `c(0.2, 0.4, 0.6, 0.8, 1)`:

- Nil (0)

- 0 - 0.2
- 0.2 - 0.4
- 0.4 - 0.6
- 0.6 - 0.8
- 0.8 - 1

Spatial reference:

This function dynamically pulls map tiles for a base map when `map = TRUE`. The inputs are projected to WGS 84/Pseudo-Mercator (EPSG:3857) to align them with the map tiles.

Zoom level:

The map tile zoom level may need to be adjusted. If the base map is blurry, increase the zoom level. Higher zoom levels will slow down the function, so only increase if necessary. Reference the OpenStreetMap Wiki for more information on zoom levels.

Value

a summary table is returned as a data frame object, Unless: `map = TRUE`: a ggplot object

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# read example area of interest geometry
geom_file_path <- "extdata/polygon_geometry.csv"
geom <- read.csv(system.file(geom_file_path, package = "fireexposuR"))

# generate an area of interest polygon with the geometry
aoi <- terra::vect(as.matrix(geom), "polygons", crs = hazard)

# generate random points within the aoi polygon
points <- terra::spatSample(aoi, 100)

# compute exposure
exposure <- fire_exp(hazard)

values_exp <- fire_exp_extract(exposure, points)

# summarize example points in a table
fire_exp_extract_vis(values_exp, classify = "local")

# visualize example points in standardized map
fire_exp_extract_vis(values_exp, map = TRUE)
```

fire_exp_map_class — *Map exposure with a classified scale*

Description

`fire_exp_map_class()` produces a standardized map by classifying exposure into predetermined exposure classes.

Usage

```
fire_exp_map_class(
  exposure,
  aoi,
  classify = c("local", "landscape", "custom"),
  class_breaks,
  zoom_level,
  title = "Classified Exposure"
)
```

Arguments

`exposure`	SpatRaster (e.g. from `fire_exp()`)
`aoi`	(Optional) SpatVector of an area of interest to mask exposure
`classify`	character, either `"local"`, `"landscape"`, or `"custom"`, to specify classification scheme to use. The default is `"local"`. If set to `"custom"`: the parameter `class_breaks` must be used.
`class_breaks`	vector of numeric values between 0-1 of the upper limits of each custom class. Ignored unless `classify = "custom"`. See details.
`zoom_level`	(Optional) numeric, set the zoom level for the basemap based on the extent of your data if defaults are not appropriate. See details. Defaults if: • `classify = "local"` or `"custom"` the zoom level default is `12` • `classify = "landscape"` the zoom level default is `7`
`title`	(Optional) String. A custom title for the plot. The default is `"Classified Exposure"`

Details

This function returns a standardized map with basic cartographic elements.

The plot is returned as a ggplot object which can be exported/saved to multiple image file formats.

This function visualizes the outputs from `fire_exp()` with classes. Classes can be chosen from the pre-set `"local"` and `"landscape"` options, or customized. To use a custom classification scheme, it should be defined with a list of numeric vectors defining the upper limits of the breaks. A Nil class is added automatically for exposure values of exactly zero.

Local classification breaks are predefined as `c(0.15, 0.3, 0.45, 1)`:

- Nil (0)
- 0 - 0.15
- 0.15 - 0.3
- 0.3 - 0.45
- 0.45 - 1

Landscape classification breaks are predefined as `c(0.2, 0.4, 0.6, 0.8, 1)`:

- Nil (0)
- 0 - 0.2
- 0.2 - 0.4
- 0.4 - 0.6
- 0.6 - 0.8
- 0.8 - 1

Spatial reference:
This function dynamically pulls map tiles for a base map. The inputs are projected to WGS 84/Pseudo-Mercator (EPSG:3857) to align them with the map tiles.

Zoom level:
The map tile zoom level may need to be adjusted. If the base map is blurry, increase the zoom level. Higher zoom levels will slow down the function, so only increase if necessary. Reference the OpenStreetMap Wiki for more information on zoom levels.

Value

a standardized map is returned as a ggplot object

See Also

`fire_exp_map_class()`

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# read example area of interest geometry
geom_file_path <- "extdata/polygon_geometry.csv"
geom <- read.csv(system.file(geom_file_path, package = "fireexposuR"))

# generate example area of interest polygon with geometry
aoi <- terra::vect(as.matrix(geom), "polygons", crs = hazard)

# compute exposure
exposure <- fire_exp(hazard)

fire_exp_map_class(exposure, aoi, classify = "local")
```

fire_exp_map_cont — *Map exposure with a continuous scale*

Description

`fire_exp_map_cont()` produces a standardized map of exposure with a continuous scale for the full extent of the data or masked to an area of interest.

Usage

```
fire_exp_map_cont(exposure, aoi, title = "Wildfire Exposure")
```

Arguments

`exposure`	SpatRaster from `fire_exp()`
`aoi`	(Optional) SpatVector of an area of interest to mask the exposure
`title`	(Optional) String. A custom title for the plot. The default is `"Wildfire Exposure"`

Details

This function returns a standardized map with basic cartographic elements. The exposure values are mapped using a continuous scale. There is no base map added with this function.

The plot is returned as a ggplot object which can be exported/saved to multiple image file formats.

Spatial Reference:

The map will be drawn using the same CRS as the input data.

Value

a map is returned as a ggplot object

See Also

`fire_exp_map_class()`

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# Compute exposure
exposure <- fire_exp(hazard)

fire_exp_map_cont(exposure)
```

fire_exp_summary *Summarize exposure by class*

Description

`fire_exp_summary()` creates a summary table of area and proportions of exposure in predetermined or custom exposure classes.

Usage

```
fire_exp_summary(
  exposure,
  aoi,
  classify = c("landscape", "local", "custom"),
  class_breaks
)
```

Arguments

`exposure`	SpatRaster from `fire_exp()`
`aoi`	(optional) SpatVector of an area of interest to mask exposure for summary
`classify`	character, either `"local"`, `"landscape"`, or `"custom"`, to specify classification scheme to use. The default is `"local"`. If set to `"custom"`: the parameter `class_breaks` must be used.
`class_breaks`	vector of numeric values between 0-1 of the upper limits of each custom class. Ignored unless `classify` = `"custom"`. See details.

Details

This function summarizes the outputs from `fire_exp()` with classes. Classes can be chosen from the pre-set `"local"` and `"landscape"` options, or customized. To use a custom classification scheme, it should be defined with a list of numeric vectors defining the upper limits of the breaks. A Nil class is added automatically for exposure values of exactly zero.

Local classification breaks are predefined as `c(0.15, 0.3, 0.45, 1)`:

- Nil (0)
- 0 - 0.15
- 0.15 - 0.3
- 0.3 - 0.45
- 0.45 - 1

Landscape classification breaks are predefined as `c(0.2, 0.4, 0.6, 0.8, 1)`:

- Nil (0)
- 0 - 0.2

- 0.2 - 0.4
- 0.4 - 0.6
- 0.6 - 0.8
- 0.8 - 1

The table reports the number of pixels, the proportion, and area in hectares and meters squared in each class.

Value

a summary table as a data frame object

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# read example area of interest polygon geometry
geom_file_path <- "extdata/polygon_geometry.csv"
geom <- read.csv(system.file(geom_file_path, package = "fireexposuR"))
aoi <- terra::vect(as.matrix(geom), "polygons", crs = hazard)

# Compute exposure
exposure <- fire_exp(hazard)

# Summary for full extent of data
fire_exp_summary(exposure, classify = "landscape")

# Summary masked to an area of interest
fire_exp_summary(exposure, aoi, classify = "landscape")
```

`fire_exp_validate` *Validate exposure with observed fires*

Description

For advanced users. `fire_exp_validate()` compares the proportion of exposure classes in a the study area to the proportion of exposure classes within observed burned areas.

Usage

```
fire_exp_validate(
  burnableexposure,
  fires,
  aoi,
  class_breaks = c(0.2, 0.4, 0.6, 0.8, 1),
  samplesize = 0.005
)
```

Arguments

burnableexposure
: A SpatRaster of exposure, non-burnable cells should be removed using optional parameter `no_burn =` in `fire_exp()`.

fires
: A SpatVector of observed fire perimeters

aoi
: (Optional) A SpatVector that delineates an area of interest

class_breaks
: (Optional) vector of numeric values between 0-1 of the upper limits of each class. The default is `c(0.2, 0.4, 0.6, 0.8, 1)`. See details.

samplesize
: Proportion of areas to sample. The default is `0.005` (0.5%)

Details

This function automates a simple validation method to assess if fire burns preferentially in areas with high exposure. The methods, and figure produced with `fire_exp_validate_plot()`, are based on Beverly et al. (2021).

The function requires an exposure raster produced for a past point in time. Cells that cannot burn, or do not represent natural land cover should be removed by setting the `no_burn` parameter in `fire_exp()` or `fire_exp_adjust()`.

The function also requires fire perimeter data. Currently, the function takes the fires as a Vector of polygons because that is typically how fire boundaries are stored in spatial databases. The fires input data should include all of the burned area that has occurred following the time period the input exposure layer was produced for. It is up to the user to determine the appropriate amount of burned area required for a meaningful assessment.

A random sample is taken to account for spatial autocorrelation, the sampled location results can be used to test for significant differences. The sample size can be adjusted. The sample size represents a proportion of cells, the default is `0.005` (0.5%). It is the user's responsibility to set an appropriate sample size.

The class breaks can be customized from the default of 0.2 intervals by setting the `class_breaks` parameter. A class of Nil is automatically added for values exactly equal to 0.

Value

a table of number of cells (n) and proportions (prop) of exposure classes within a sampled area (Sample) and across the full extent (Total).for the full extent of the exposure data (expected) and only within the burned areas (observed).

References

Beverly JL, McLoughlin N, Chapman E (2021) A simple metric of landscape fire exposure. *Landscape Ecology* **36**, 785-801. doi:10.1007/s10980020011738

See Also

`fire_exp_validate_plot()`

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# generate example non-burnable cells data
geom_file_path <- "extdata/polygon_geometry.csv"
geom <- read.csv(system.file(geom_file_path, package = "fireexposuR"))
polygon <- terra::vect(as.matrix(geom), "polygons", crs = hazard)
no_burn <- terra::rasterize(polygon, hazard)

# generate example fire polygons by buffering random points
points <- terra::spatSample(terra::rescale(hazard, 0.8),
                            30, as.points = TRUE)
fires <- terra::buffer(points, 800)
# PLEASE NOTE THIS EXAMPLE DATA DOES NOT GENERATE MEANINGFUL RESULTS

# compute exposure and remove non-burnable cells
exposure <- fire_exp(hazard, no_burn = no_burn)

# validation table
fire_exp_validate(exposure, fires)
```

`fire_exp_validate_plot` *Visualize validation outputs in a plot*

Description

`fire_exp_validate_plot()` Visualizes the results from `fire_exp_validate()` in a bar plot

Usage

```
fire_exp_validate_plot(
  validation_table,
  what = c("both", "total", "sample"),
  title
)
```

Arguments

`validation_table`	The output table from `fire_exp_validate()`
`what`	string. Which plot should be returned? Can be `sample`, `total`, or `"both"` (Default)
`title`	Optional. String. Add a custom title to the plot.

Value

a ggplot object

See Also

fire_exp_validate()

Examples

```
# read example hazard data
hazard_file_path <- "extdata/hazard.tif"
hazard <- terra::rast(system.file(hazard_file_path, package = "fireexposuR"))

# generate example non-burnable cells data
geom_file_path <- "extdata/polygon_geometry.csv"
geom <- read.csv(system.file(geom_file_path, package = "fireexposuR"))
polygon <- terra::vect(as.matrix(geom), "polygons", crs = hazard)
no_burn <- terra::rasterize(polygon, hazard)

# generate example fire polygons by buffering random points
points <- terra::spatSample(terra::rescale(hazard, 0.8),
                            30, as.points = TRUE)
fires <- terra::buffer(points, 800)
# PLEASE NOTE THIS EXAMPLE DATA DOES NOT GENERATE MEANINGFUL RESULTS

# compute exposure and remove non-burnable cells
exposure <- fire_exp(hazard, no_burn = no_burn)

# results as table
validation_outputs <- fire_exp_validate(exposure, fires)

# results as bar chart
fire_exp_validate_plot(validation_outputs)
```

Index