

Package ‘Rothermel’

September 10, 2026

Type Package

Title Surface Fire Spread Model

Version 1.4.1

Date 2026-09-10

Description Implements the surface fire spread model of Rothermel (1972) <[doi:10.2737/INT-RP-115](https://doi.org/10.2737/INT-RP-115)> in R. Additional utilities support uncertainty propagation, selection among standard fuel models, optimization of fuel model parameters by genetic algorithms, and example datasets.

License GPL-2

Encoding UTF-8

Depends R (>= 3.5.0)

Imports GA (>= 3.2.0), ftsa (>= 3.1)

Suggests spelling

Language en-US

NeedsCompilation no

Author Giorgio Vacchiano [aut, cre] (ORCID: <<https://orcid.org/0000-0001-8100-0659>>),
Davide Ascoli [ctb] (ORCID: <<https://orcid.org/0000-0002-2671-5122>>)

Maintainer Giorgio Vacchiano <gvacchiano@gmail.com>

Repository CRAN

Date/Publication 2026-09-10 16:50:13 UTC

Contents

bestFM	2
firexp	3
gaRoth	5
ros	8
rosunc	12
Rothermel	14
scenarios	15
SFM_metric	16

bestFM	<i>Function to select best standard fuel model for observed fire rate of spread</i>
--------	---

Description

The function preloads the 13 fire behavior fuel models by Albini (1976) and the 40 fuel models by Scott & Burgan (2005), computes rate of spread using Rothermel’s (1972) model for a vector or data frame of fire experiment data, and computes root mean square error and mean bias of each fuel model to observed rate of spread.

Usage

bestFM(obs, m, u, slope)

Arguments

obs	a single value, or vector, of observed ROS [m/min].
m	a vector, or data frame, of percent moisture on a dry weight basis [percent] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent).
u	a single value, or vector, of midflame windspeed [km/h].
slope	a single value, or vector, of percent site slope [percent].

Value

If a single observation of rate of spread is analyzed (length of obs =1), a vector of bias values, computed as (predicted - observed ROS) for every standard fire behavior fuel model. If more observations are analyzed (length of obs >1), a data.frame of root mean square error and mean bias computed from observed and predicted ROS using every standard fire behavior fuel model.

Author(s)

Giorgio Vacchiano, Davide Ascoli (DISAFA, University of Torino, Italy)

References

Albini, F. A. (1976). Estimating wildfire behavior and effects. Gen. Tech. Rep. INT-30. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

Rothermel, R. C. (1972). A mathematical model for fire spread predictions in wildland fires. Research Paper INT-115. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

Scott, J., Burgan, R. E. (2005). A new set of standard fire behavior fuel models for use with Rothermel’s surface fire spread model. Gen. Tech. Rep. RMRS-GTR-153. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

See Also[ros, SFM_metric](#)**Examples**

```
data ("firexp")
a <- bestFM (obs = firexp$ros,
             m = firexp[,18:22],
             u = firexp[, "u"],
             slope = firexp[, "slope"])

a

# plot results
barplot (a$error,
        cex.names = 0.7,
        las = 2,
        ylab = "rmse",
        col = sign (a$bias) + 3,
        names.arg = rownames (a))

legend (1, max (a [, 1]),
        c ("Overprediction", "Underprediction"),
        col = c ("blue", "red"),
        pch = 15)
```

firexp

*field fire experiments in Vauda Nature Reserve, Italy***Description**

The dataset include observations of surface head fire rate of spread (ROS) measured during field fire experiments in heathland fuels (mixed grass-shrub) carried out on flat terrain under differing fire weather conditions (Ascoli et al. 2009). For each ROS values of Rothermel's fire behavior fuel models parameters and environmental conditions are provided. Some ranges in the dataset are: ROS <- 0.9 - 26.3 m/min; wind speed <- 0.4 - 7.9 km/h; 1h fuels moisture: 10 - 27 percent.

Usage

```
data (firexp)
```

Format

A data frame with 40 observations on the following 25 variables.

Load_1h loading of 1h fuel class [t/ha]

Load_10h loading of 10h fuel class [t/ha]

Load_100h loading of 100h fuel class [t/ha]

Load_Live_Herb loading of herbaceous fuels [t/ha]

Load_Live_Woody loading of woody fuels [t/ha]
 SA/V_1h SAV of 1h fuel class [m2/m3]
 SA/V_10h SAV of 10h fuel class [m2/m3]
 SA/V_100h SAV of 100h fuel class [m2/m3]
 SA/V_Live_Herb SAV of herbaceous fuels [m2/m3]
 SA/V_Live_Woody SAV of woody fuels [m2/m3]
 Fuel_Bed_Depth fuel bed depth [cm]
 Mx_dead dead fuel moisture of extinction [percent]
 Heat_1h heat content of 1h fuel class [kJ/kg]
 Heat_10h heat content of 10h fuel class [kJ/kg]
 Heat_100h heat content of 100h fuel class [kJ/kg]
 Heat_Live_Herb heat content of herbaceous fuels [kJ/kg]
 Heat_Live_Woody heat content of woody fuels [kJ/kg]
 Moist_1h moisture of 1h fuel class on a dry weight basis [percent]
 Moist_10h moisture of 10h fuel class on a dry weight basis [percent]
 Moist_100h moisture of 100h fuel class on a dry weight basis [percent]
 Moist_Live_Herb moisture of herbaceous fuels on a dry weight basis [percent]
 Moist_Live_Woody moisture of woody fuels on a dry weight basis [percent]
 u midflame windspeed [km/h]
 slope site slope [
 ros observed rate of spread [m/min]]

Source

Ascoli, D., Marzano, R., and Bovio, G. (2006). Experimental fires for heather moorland management in north-western Italy. *Forest Ecology and Management* 234S: 258. <https://doi.org/10.1016/j.foreco.2006.08.287>.

References

Ascoli, D., Beghin, R., Ceccato, R., Gorlier, A., Lombardi, G., Lonati, M., Marzano, R., Bovio, G., and Cavallero, A. (2009). Developing an adaptive management approach to prescribed burning: a long-term heathland conservation experiment in north-west Italy. *International Journal of Wildland Fire* 18(6): 727-735. <https://doi.org/10.1071/WF07114>.

Examples

```
data (firexp)
boxplot (firexp$ros)
```

Description

Genetic algorithms (GA) are a technique of machine-based mathematical optimization. The algorithm searches, within user-defined ranges, for values that minimize or maximize a target function. Here, fuel model parameters are searched that minimize root mean square error (RMSE) of forward fire rate of spread predicted by Rothermel (1972) model against observed data. Depends on package "GA" (Scrucca 2013) for the execution of the genetic algorithm; refer to this publication for a full explanation of GA parameters and settings.

Usage

```
gaRoth (
  w_1h = range(SFM_metric[, -1][, 1]),
  w_10h = range(SFM_metric[, -1][, 2]) ,
  w_100h = range(SFM_metric[, -1][, 3]) ,
  w_Live_Herb = range(SFM_metric[, -1][, 4]) ,
  w_Live_Woody = range(SFM_metric[, -1][, 5]) ,
  s_1h = range(SFM_metric[, -1][, 6]) ,
  s_10h = range(SFM_metric[, -1][, 7]) ,
  s_100h = range(SFM_metric[, -1][, 8]) ,
  s_Live_Herb = range(SFM_metric[, -1][, 9]) ,
  s_Live_Woody = range(SFM_metric[, -1][, 10]) ,
  delta = range(SFM_metric[, -1][, 11]) ,
  mx.dead = range(SFM_metric[, -1][, 12]) ,
  h_1h = range(SFM_metric[, -1][, 13]) ,
  h_10h = range(SFM_metric[, -1][, 14]) ,
  h_100h = range(SFM_metric[, -1][, 15]) ,
  h_Live_Herb = range(SFM_metric[, -1][, 16]) ,
  h_Live_Woody = range(SFM_metric[, -1][, 17]) ,
  m, u, slope, modeltype, obs,
  method="rmse", maxiter=50, popSize = 20, pcrossover = 0.8,
  pmutation = 0.1, elitism = base::max(1, round(popSize * 0.05)),
  ...)
```

Arguments

w_1h	a vector of minimum and maximum fuel load [t/ha] for fuel class 1-hour; (0,0) if fuel class is absent.
w_10h	a vector of minimum and maximum fuel load [t/ha] for fuel class 10-hour; (0,0) if fuel class is absent.
w_100h	a vector of minimum and maximum fuel load [t/ha] for fuel class 100-hour; (0,0) if fuel class is absent.

w_Live_Herb	a vector of minimum and maximum fuel load [t/ha] for live herbs; (0,0) if fuel class is absent.
w_Live_Woody	a vector of minimum and maximum fuel load [t/ha] for live woody fuels; (0,0) if fuel class is absent.
s_1h	a vector of minimum and maximum surface-to-volume ratio [m ² /m ³] for fuel class 1-hour; (0,0) if fuel class is absent.
s_10h	a vector of minimum and maximum surface-to-volume ratio [m ² /m ³] for fuel class 10-hour; (0,0) if fuel class is absent.
s_100h	a vector of minimum and maximum surface-to-volume ratio [m ² /m ³] for fuel class 100-hour; (0,0) if fuel class is absent.
s_Live_Herb	a vector of minimum and maximum surface-to-volume ratio [m ² /m ³] for live herbs; (0,0) if fuel class is absent.
s_Live_Woody	a vector of minimum and maximum surface-to-volume ratio [m ² /m ³] for live woody fuels; (0,0) if fuel class is absent.
delta	a vector of minimum and maximum fuel bed depth [cm].
mx.dead	a vector of minimum and maximum dead fuel moisture of extinction [percent].
h_1h	a vector of minimum and maximum heat content [kJ/kg] for fuel class 1-hour; (0,0) if fuel class is absent.
h_10h	a vector of minimum and maximum heat content [kJ/kg] for fuel class 10-hour; (0,0) if fuel class is absent.
h_100h	a vector of minimum and maximum heat content [kJ/kg] for fuel class 100-hour; (0,0) if fuel class is absent.
h_Live_Herb	a vector of minimum and maximum heat content [kJ/kg] for live herbs; (0,0) if fuel class is absent.
h_Live_Woody	a vector of minimum and maximum heat content [kJ/kg] for live woody fuels; (0,0) if fuel class is absent.
modeltype	select S(tatic) or D(ynamic) Rothermel model.
obs	a vector of observed ROS [m/min].
m	a data frame of percent moisture on a dry weight basis [percent] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent).
u	a vector of midflame windspeed [km/h].
slope	a vector of percent site slope [percent].
method	method of forecast error measure, see error .
maxiter	the maximum number of iterations to run before the GA search is halted.
popSize	the population size.
pcrossover	the probability of crossover between pairs of chromosomes. Typically this is a large value and by default is set to 0.8.
pmutation	the probability of mutation in a parent chromosome. Usually mutation occurs with a small probability, and by default is set to 0.1.
elitism	the number of best fitness individuals to survive at each generation. By default the top 5/100 individuals will survive at each iteration.
...	additional arguments to be passed to ga .

Details

The function defaults to a real-valued coded GA. Default values for fuel models are currently extracted from the standard Fuel Models table by Anderson (1982) and Scott and Burgan (2005). GA optimization is calculated from a vector of predicted vs. observed data; therefore, vectors of two or more values must be entered for observations, wind speed, and slope, and a data frame with 5 columns and two or more lines must be entered for fuel moisture (see [ros](#)). Since [ga](#) defaults to maximizing the fitness function, the opposite of the error metric is computed (i.e., with a negative sign).

Value

Returns an object of class [ga-class](#). The best solution is reported by the S4 slot "solution". The opposite of the fitness value is reported by the slot "fitnessValue".

Author(s)

Giorgio Vacchiano, Davide Ascoli (DISAFA, University of Torino, Italy)

References

Rothermel, R. C. (1972). A mathematical model for fire spread predictions in wildland fires. Research Paper INT-115. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

Scrucca, L. (2013). GA: A Package for Genetic Algorithms in R. Journal of Statistical Software, 53(4), 1-37. DOI [doi:10.18637/jss.v053.i04](https://doi.org/10.18637/jss.v053.i04).

See Also

[ros](#), [fireexp](#), [SFM_metric](#)

Examples

```
library (Rothermel)

# Observed site variables and ROS
data (fireexp)
m <- fireexp [, 18:22]
u <- fireexp [, "u"]
slope <- fireexp [, "slope"]
obs <- fireexp [, "ros"]
modeltype <- "S"

# GA optimization using minimization of Root mean square error ("rmse")
# Ranges for fuel model parameters after Ascoli et al. (2015)

ga.model <- gaRoth (m=m, u=u, slope=slope, obs=obs, modeltype=modeltype,
  method="rmse", popSize=10, maxiter=2, seed=1,
  w_1h = c(1.5,7.24),
  w_10h = c(0,0),
  w_100h = c(0,0),
```

```

w_Live_Herb = c(0,0),
w_Live_Woody = c(2.9,10.4),
s_1h = c(6640,10036),
s_10h = c(0,0),
s_100h = c(0,0),
s_Live_Herb = c(0,0),
s_Live_Woody = c(8810,10560),
delta = c(19,70),
mx.dead = c(25,50),
h_1h = c(18719,19919),
h_10h = c(0,0),
h_100h = c(0,0),
h_Live_Herb = c(0,0),
h_Live_Woody = c(20000,22504)
)

summary(ga.model)

```

ros	<i>Function to predict Rothermel's (1972) rate of spread [m/min] for surface headfires</i>
-----	--

Description

Include corrections to the original model by Frandsen (1973), Albini (1976), and Andrews et al. (2013).

Usage

```
ros (modeltype, w, s, delta, mx.dead, h, m, u, slope)
```

Arguments

modeltype	S(tatic), D(ynamic)
w	a vector or data frame of fuel load [t/ha] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent).
s	a vector or data frame of surface-to-volume ratio [m ² /m ³] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent).
delta	a value or vector of fuel bed depth [cm]
mx.dead	a value or vector of dead fuel moisture of extinction [percent]
h	a vector or data frame of heat content [kJ/kg] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent).

m	a vector or data frame of percent moisture on a dry weight basis (percent) for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values or columns; 0 if fuel class is absent).
u	a value or vector of midflame windspeed [km/h]
slope	a value or vector of site slope [percent]

Value

A list of values or vectors for the following variables: [1] Characteristic dead fuel moisture [percent], [2] Characteristic live fuel moisture [percent], [3] Live fuel moisture of extinction [percent], [4] Characteristic SAV [m²/m³] [5] Bulk density [kg/m³], [6] Packing ratio [dimensionless], [7] Relative packing ratio [dimensionless], [8] Dead fuel Reaction intensity [kW/m²], [9] Live fuel Reaction intensity [kW/m²], [10] Reaction intensity [kW/m²], [11] Wind factor [0-100], [12] Slope factor [0-1], [13] Heat source [kW/m²], [14] Heat sink [kJ/m³], [15] ROS [m/min].

Author(s)

Giorgio Vacchiano, Davide Ascoli (DISAFA, University of Torino, Italy)

References

- Albini, F. A. (1976). Computer-based models of wildland fire behavior: A users' manual. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.
- Andrews, P. L., Cruz, M. G., and Rothermel, R. C. (2013). Examination of the wind speed limit function in the Rothermel surface fire spread model. *International Journal of Wildland Fire* 22 (7): 959-969. <https://doi.org/10.1071/WF12122>.
- Frandsen, W. H. (1973). Using the effective heating number as a weighting factor in Rothermel's fire spread model. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.
- Rothermel, R. C. (1972). A mathematical model for fire spread predictions in wildland fires. Research Paper INT-115. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

See Also

[rosunc](#), [SFM_metric](#), [scenarios](#)

Examples

```
# Example 1: Simulation using vectors of input values

modeltype <- "D"
w <-c (2, 1, 0.5, 3, 8)
s <- c (5600, 358, 98, 6200, 8000)
delta <- 50
mx.dead <- 30
h <- c (18622, 18622, 18622, 19500, 20000)
m <- c (7, 8, 9, 40, 60)
```

```

u <- 5
slope <- 10

ros(modeltype, w, s, delta, mx.dead, h, m, u, slope)

# Example 2: variable wind input
# Only rate of spread is reported here (i.e., element [15] of ros ( ) output)

modeltype <- "D"
w <-c (2, 1, 0.5, 3, 8)
s <- c (5600, 358, 98, 6200, 8000)
delta <- 50
mx.dead <- 30
h <- c (18622, 18622, 18622, 19500, 20000)
m <- c (7, 8, 9, 40, 60)
slope <- 10

df <- data.frame ("windspeed" = seq (3, 15, 1), ROS=NA)

for (i in 1:nrow (df)) {
  df [i,2] <-
    ros (modeltype, w, s, delta, mx.dead, h, m, u=df[i,1], slope) [15]
}
df

# Example 3: variable wind and slope input
# A two-entry table of rates of spread is created

modeltype <- "D"
w <-c (2, 1, 0.5, 3, 8)
s <- c (5600, 358, 98, 6200, 8000)
delta <- 50
mx.dead <- 30
h <- c (18622, 18622, 18622, 19500, 20000)
m <- c (7, 8, 9, 40, 60)

u <- seq (3, 15, 1)
slope <- seq (0, 45, 15)

df <- matrix (rep (NA, length (u) * length (slope)),
              length (u),
              length (slope)
              )

df <- data.frame (u, df)
colnames (df) <- c ("windspeed",
                   paste ("slope_", as.character (slope))
                   )

for (i in 1:length (u)) {
  for (j in 1:length (slope)) {
    df [i, j+1] <- ros (
      modeltype, w, s, delta, mx.dead, h, m, u[i], slope[j])[15]
  }
}

```

```

    }
  }

  df

# Example 4: prediction and validation of rate of spread
# using existing data from a field experiment

library (Rothermel)

# Observed variables
data (firexp)
m <- firexp [, 18:22]
u <- firexp [, "u"]
slope <- firexp [, "slope"]
obs <- firexp [, "ros"]

# Predict ROS using Standard Fuel Models GR5, GS3 and SH7
data (SFM_metric)
a = list ( )
models = which (rownames (SFM_metric) == "GR5" |
               rownames (SFM_metric) == "GS3" |
               rownames (SFM_metric) == "SH7")
for (i in 1 : length (models) ) {
  modeltype <- SFM_metric [models [i], 1]
  w <- SFM_metric [models [i], 2:6]
  s <- SFM_metric [models [i], 7:11]
  delta <- SFM_metric [models [i], "Fuel_Bed_Depth"]
  mx.dead <- SFM_metric [models [i], "Mx_dead"]
  h <- SFM_metric [models [i], 14:18]
  a [i] <- ros (modeltype, w, s, delta, mx.dead, h,
               m, u, slope)[15]}

# Plot

oldpar <- par(mfrow = c(1,1))

plot (obs, a [[1]], xlab = "Observed rate of spread (m/min)",
      ylab = "Predicted rate of spread (m/min)", col = "red",
      pch=19, xlim = c (0, 30), cex.lab = 1.1)
points (obs, a [[2]], pch = 19, col = "green2")
points (obs, a [[3]], pch = 19, col = "blue2")
abline (coef = c(0, 1))
abline (coef = c(0, 0.7),lty = 2); text (13.6, 19.2, "-30 percent")
abline (coef = c(0, 1.3),lty = 2); text (28.7, 19.2, "+30 percent")
legend (0, 19.2, c("GR5", "GS3", "SH7"), pch = 19,
       col = c("red", "green2", "blue2"), title = "Fuel model")

# Inset Residual plot (not run)
par (fig = c (.57, .98, .07, .55), new = TRUE)
plot (obs, a[[1]] - obs, xlab= "", ylab= "", col = "red",
     main= "Residuals", font.main = 1, pch=19, cex=.7)
points (obs, a [[2]] - obs, pch = 19, cex =.7, col = "green2")

```

```

points (obs, a [[3]] - obs, pch = 19, cex = .7, col = "blue2")
abline (h = 0)
par (fig = c (0, 1, 0, 1))
par(oldpar)

```

rosunc

Function to carry out uncertainty propagation analysis on Rothermel's (1972) fire spread model

Description

This function offers the user the possibility to perturb inputs to Rothermel's (1972) fire behavior model and propagate the uncertainty to the resulting estimate of Rate of spread [m/min] by means of Monte Carlo iterative sampling. Random values are extracted from Gaussian distributions with mean = observed values, and spread defined by a custom ratio of standard deviation to the mean defined by the user.

Usage

```

rosunc (modeltype, w, s, delta, mx.dead, h, m, u, slope,
        sdu = 0, sdm = 0, sds = 0, sdw = 0, sdd = 0,
        nsim = 1000)

```

Arguments

modeltype	S(tatic), D(ynamic)
w	a vector of fuel load [t/ha] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values; 0 if fuel class is absent).
s	a vector of surface-to-volume ratio [m ² /m ³] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values; 0 if fuel class is absent).
delta	atomic vector, fuel bed depth [cm]
mx.dead	atomic vector, dead fuel moisture of extinction [percent]
h	a vector of heat content [kJ/kg] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values; 0 if fuel class is absent).
m	a vector of percent moisture on a dry weight basis [percent] for fuel classes 1-hour, 10-hour, 100-hour, live herbs and live woody, respectively (5 values; 0 if fuel class is absent).
u	atomic vector, midflame windspeed [km/h]
slope	atomic vector, site slope [percent]
sdu	coefficient of variation for wind speed (ratio of standard deviation to the mean; default = no perturbation)
sdm	coefficient of variation for fuel moistures (ratio of standard deviation to the mean; default = no perturbation)

sds	coefficient of variation for slope (ratio of standard deviation to the mean; default = no perturbation)
sdw	coefficient of variation for fuel loadings (ratio of standard deviation to the mean; default = no perturbation)
sdd	coefficient of variation for fuel bed depth (ratio of standard deviation to the mean; default = no perturbation)
nsim	number of Monte Carlo iterations (default =1000)

Value

A vector of predicted ROS [m/min] from Monte Carlo simulations.

Author(s)

Giorgio Vacchiano, Davide Ascoli (DISAFA, University of Torino, Italy)

References

- Cruz M. G. (2010). Monte Carlo-based ensemble method for prediction of grassland fire spread. *International Journal of Wildland Fire* 19: 521-530.
- Jimenez E., Hussaini M. Y., Goodrick S. (2008). Quantifying parametric uncertainty in the Rothermel model. *International Journal of Wildland Fire*, 17: 638-649.
- Rothermel, R. C. (1972). A mathematical model for fire spread predictions in wildland fires. Research Paper INT-115. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

See Also

[ros](#), [SFM_metric](#), [firexp](#)

Examples

```
data ("firexp")
varnames <- names (firexp)

# select only one observation and create a numeric vector for function input
firexp <- as.numeric (firexp [5,])
names (firexp) <- varnames

pred <- rosunc (
  modeltype = "D",
  w = firexp [1:5],
  s = firexp [6:10],
  delta = firexp ["Fuel_Bed_Depth"],
  mx.dead = firexp ["Mx_dead"],
  h = firexp [13:17],
  m = firexp [18:22],
  u = firexp ["u"],
  slope = firexp ["slope"],
  sdm = 0.3,
```

```

      nsim = 100)

summary (pred)

# Figure
hist (pred,
      xlab = "ROS [m/min]",
      freq = FALSE,
      xlim = c (0, max (pred)),
      breaks = 20,
      main = "")
lines (density (pred), lty=2, lwd=2)
abline (v = firexp ["ros"], col = "red")
text (firexp ["ros"],
      max (density (pred)$y),
      labels = "obs",
      pos = 4)

```

Rothermel

Rothermel surface fire spread model for R

Description

R build of Rothermel's (1972) model for surface head fire rate of spread with some additional utilities (best standard fuel model selection, uncertainty propagation, optimization of fuel models by genetic algorithms) and sample datasets.

Details

Arguments to `ros()`, `rosunc()`, and `bestFM()` should be provided by the user from standard or custom fire behavior fuel models, standard or custom moisture scenarios, and desired wind speed and slope. `rosunc()` works with individual observations; `ros()` and `bestFM()` work with individual or multiple observations.

Author(s)

Giorgio Vacchiano (University of Milan, Italy) and Davide Ascoli (University of Torino, Italy)

References

- Rothermel, R. C. (1972). A mathematical model for fire spread predictions in wildland fires. Research Paper INT-115. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.
- Vacchiano, G., Ascoli, D. (2015). An implementation of the Rothermel fire spread model in the R programming language. *Fire Technology* 51: 523–535. DOI [doi:10.1007/s1069401404056](https://doi.org/10.1007/s1069401404056).

See Also

[ros](#), [rosunc](#), [bestFM](#), [gaRoth](#)

scenarios

Moisture scenarios by Scott & Burgan (2005)

Description

A fuel Moisture Scenario is a set of fuel moisture values representing moisture condition of the surface fuel. Fuel Moisture Scenarios are useful for testing custom Fuel Models or developing fire prescriptions. For example, a moisture Scenario might be developed to represent a location's 90th-, 95th-, and 97th-percentile fire weather conditions (Andrews 2009)

Usage

```
data(scenarios)
```

Format

A data frame with 16 observations on the following 12 variables.

Moist_1h moisture of 1h fuel class on a dry weight basis [percent]

Moist_10h moisture of 10h fuel class on a dry weight basis [percent]

Moist_100h moisture of 100h fuel class on a dry weight basis [percent]

Moist_Live_Herb moisture of herbaceous fuels on a dry weight basis [percent]

Moist_Live_Woody moisture of woody fuels on a dry weight basis [percent]

Description Scenario description

Source

Andrews, P. L. (2009). BehavePlus fire modeling system, version 5.0: Variables. Gen. Tech. Rep. RMRS-GTR-213WWW Revised. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station. <https://doi.org/10.2737/RMRS-GTR-213>

Scott, J., & Burgan, R. E. (2005). A new set of standard fire behavior fuel models for use with Rothermel's surface fire spread model. Gen. Tech. Rep. RMRS-GTR-153. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Examples

```
data(scenarios)
barplot(as.matrix(scenarios[1, 1:5]),
        cex.names = .6,
        main = rownames(scenarios)[1])

# ROS using Standard fuel models and Moisture scenarios
data(scenarios, SFM_metric)
modeltype <- SFM_metric["GR1", "Fuel_Model_Type"]
w <- SFM_metric["GR1", 2:6]
s <- SFM_metric["GR1", 7:11]
delta <- SFM_metric["GR1", "Fuel_Bed_Depth"]
```

```

mx.dead <- SFM_metric ["GR1", "Mx_dead"]
h <- SFM_metric ["GR1", 14:18]
m <- scenarios ["D2L2", 1:5]
u <- 8
slope <- 0

a <- ros (modeltype, w, s, delta, mx.dead, h, m, u, slope)
a

```

SFM_metric

*Standard fuel models (Albini 1976, Scott & Burgan 2005)***Description**

All values converted to metric format.

Usage

```
data(SFM_metric)
```

Format

A data frame including standard fire behavior fuel models as in Albini (1976) and Scott and Burgan (2005) to be used as input of `ros ( )` function. All values converted to metric format.

Fuel_Model_Type S(tatic), D(ynamic)
 Load_1h loading of 1h fuel class [t/ha]
 Load_10h loading of 10h fuel class [t/ha]
 Load_100h loading of 100h fuel class [t/ha]
 Load_Live_Herb loading of herbaceous fuels [t/ha]
 Load_Live_Woody loading of woody fuels [t/ha]
 SA/V_1h SAV of 1h fuel class [m2/m3]
 SA/V_10h SAV of 10h fuel class [m2/m3]
 SA/V_100h SAV of 100h fuel class [m2/m3]
 SA/V_Live_Herb SAV of herbaceous fuels [m2/m3]
 SA/V_Live_Woody SAV of woody fuels [m2/m3]
 Fuel_Bed_Depth fuel bed depth [cm]
 Mx_dead dead fuel moisture of extinction [percent]
 Heat_1h heat content of 1h fuel class [kJ/kg]
 Heat_10h heat content of 10h fuel class [kJ/kg]
 Heat_100h heat content of 100h fuel class [kJ/kg]
 Heat_Live_Herb heat content of herbaceous fuels [kJ/kg]
 Heat_Live_Woody heat content of woody fuels [kJ/kg]

Source

Albini, F. A. (1976). Computer-based models of wildland fire behavior: A users' manual. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

Scott, J., & Burgan, R. E. (2005). A new set of standard fire behavior fuel models for use with Rothermel's surface fire spread model. Gen. Tech. Rep. RMRS-GTR-153. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Examples

```
data(SFM_metric)

# ROS using Standard fuel models and Moisture scenarios
data (scenarios, SFM_metric)
modeltype <- SFM_metric ["GR1", "Fuel_Model_Type"]
w <- SFM_metric ["GR1", 2:6]
s <- SFM_metric ["GR1", 7:11]
delta <- SFM_metric ["GR1", "Fuel_Bed_Depth"]
mx.dead <- SFM_metric ["GR1", "Mx_dead"]
h <- SFM_metric ["GR1", 14:18]
m <- scenarios ["D2L2", 1:5]
u <- 8
slope <- 0

a <- ros (modeltype, w, s, delta, mx.dead, h, m, u, slope)
a
```

Index

- * **datasets**
 - firexp, [3](#)
 - scenarios, [15](#)
 - SFM_metric, [16](#)
- * **models**
 - bestFM, [2](#)
 - rosunc, [12](#)
- * **model**
 - gaRoth, [5](#)
 - ros, [8](#)
- * **package**
 - Rothermel, [14](#)

bestFM, [2](#), [14](#)

error, [6](#)

firexp, [3](#), [7](#), [13](#)

ga, [6](#), [7](#)

gaRoth, [5](#), [14](#)

ros, [3](#), [7](#), [8](#), [13](#), [14](#)

rosunc, [9](#), [12](#), [14](#)

Rothermel, [14](#)

scenarios, [9](#), [15](#)

SFM_metric, [3](#), [7](#), [9](#), [13](#), [16](#)