Package ‘viabilitymetrics’

March 30, 2018

Type Package

Title Seed Viability Calculations and Curve Fitting

Version 0.0.0.9000

Description An implementation of viability equation of Ellis and Roberts (1980) for calculation of several seed viability metrics such as storage period, final viability, storage moisture content, storage temperature and days to loose one probit viablity. The package further includes various conversions and transformations associated with seed viability calculations.

Copyright 2017-2018, ICAR-NBPGR

License GPL-2 | GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.0.1)

VignetteBuilder knitr

RoxygenNote 6.0.1

Imports data.table,
ggplot2,
plyr,
Rdpack

Suggests knitr

RdMacros Rdpack

URL https://github.com/aravind-j/viabilitymetrics,
https://aravind-j.github.io/viabilitymetrics/

BugReports https://github.com/aravind-j/viabilitymetrics/issues

R topics documented:

`dry2wet` *Conversion between Moisture content (wet/fresh weight basis) and Moisture content (dry weight basis)*

Description

These functions convert from moisture content determined on wet/fresh weight basis to equivalent value on dry weight basis and back. `MoistureNomograph` plots the nomograph for these conversions.

Usage

```
dry2wet(mc)

wet2dry(mc)

MoistureNomograph(min, max, basis = c("wet", "dry"), horiz = FALSE)
```

Arguments

`mc`	Moisture content.
`min`	Minimum value of moisture content to be plotted in nomograph.
`max`	Minimum value of moisture content to be plotted in nomograph.
`basis`	The basis on which moisture content is estimated
`horiz`	If `TRUE`, nomograph is plotted horizontally.

Details

Conversions between moisture content (%) determined on wet weight basis MC_{wb} and that on dry weight basis MC_{db} are computed based on the formulae of Cromarty et al., (1982) as follows:

$$MC_{db} = \frac{100 \times MC_{wb}}{100 - MC_{wb}}$$

$$MC_{wb} = \frac{100 \times MC_{db}}{100 + MC_{db}}$$

$$MC_{wb} = \frac{w_i - w_f}{w_i}$$

$$MC_{db} = \frac{w_i - w_f}{w_f}$$

Where, w_i is the initial weight and w_f is the final weight.

If the moisture content (`mc`) for conversion is beyond limits (0-50 % for `wet2dry` and 0-100 % for `dry2wet`), a warning is issued.

Value

For wet2dry and dry2wet, the converted moisture content (%).

For MoistureNomograph, the nomograph as an object of class ggplot.

References

Cromarty A, Ellis RH and Roberts EH (1982). *The Design of Seed Storage Facilities for Genetic Conservation*, Revised 1985 and 1990 edition. International Board for Plant Genetic Resources, Rome, Italy.

Examples

```
#----------------------------------------------------------------------------
# Moisture content (wet basis) to moisture content (dry basis)
#----------------------------------------------------------------------------
wet2dry(25)
# Warning if moisture content is beyond limits (0-50 %)
wet2dry(60)
wet2dry(-10)

#----------------------------------------------------------------------------
# Moisture content (dry basis) to moisture content (wet basis)
#----------------------------------------------------------------------------
dry2wet(30)
# Warning if moisture content is beyond limits (0-100 %)
dry2wet(-10)
dry2wet(110)

#----------------------------------------------------------------------------
# Nomograph
#----------------------------------------------------------------------------
# Horizontal
MoistureNomograph(min = 0, max = 50, basis = "wet", horiz = TRUE)
MoistureNomograph(min = 0, max = 100, basis = "dry", horiz = TRUE)

# Vertical
MoistureNomograph(min = 0, max = 50, basis = "wet", horiz = FALSE)
MoistureNomograph(min = 0, max = 100, basis = "dry", horiz = FALSE)

# Nomograph is a "ggplot" object
nom <- MoistureNomograph(min = 0, max = 50, basis = "wet", horiz = TRUE)
library(ggplot2)
nom + geom_hline(aes(yintercept=30), colour = "red")
p <- "Scale for converting moisture content values\nbetween dry and wet basis"
cap <- "based on Cromarty et al., 1982"
nom + labs(title  = p, caption = cap) +
  theme(plot.title = element_text(hjust = 0.5))
```

FinalViability *Predict variables from the seed viability equation*

Description

Predict the variables, final viability, storage period, storage temperature and storage period from the improved seed viability equation (Ellis and Roberts, 1980).

`FinalViability` Compute the final viability after a period of storage at a set of storage conditions (seed moisture content and temperature).

`StoragePeriod` Compute the storage period from the final viability and the storage conditions (seed moisture content and temperature).

`StorageMC` Compute the storage moisture content to give final viability at a particular storage temperature.

`StorageTemp` Compute the storage temperature to give final viability at a particular storage temperature.

Usage

```
FinalViability(initial, period, vcindex, vcdirect, mc, temp, years = FALSE)

StorageMC(initial, final, period, vcindex, vcdirect, temp, years = FALSE)

StorageTemp(initial, final, period, vcindex, vcdirect, mc, years = FALSE,
  unit = c("celsius", "fahrenheit"))

StoragePeriod(initial, final, vcindex, vcdirect, mc, temp, years = FALSE)
```

Arguments

`initial` The initial viability (%).

`period` The time period of storage in days or years according to the argument `years`).

`vcindex` An interger value indicating the index of seed viability. constants to be used from the `viabilityconstants` dataset in the package.

`vcdirect` A numeric vector of length 4 with the four viability constants *viz.*: K_E, C_W, C_H and C_Q.

`mc` Moisture content.

`temp` Temperature in °C.

`years` If `TRUE`, returns the output period in years instead of days.

`final` The final viability (%).

`unit` The unit of temperature.

Details

The improved seed viability equation of Ellis and Roberts (1980) describes the relationship between final viability, storage period and storage environment conditions as follows:

$$v = K_i - \frac{p}{\sigma}$$

or

$$v = K_i - \left(\frac{1}{\sigma}\right) \cdot p$$

Where, v is the probit percentage viability at storage time p (final viability), K_i is the probit percentage viability of the seedlot at the beginning of storage (seedlot constant) and $\frac{1}{\sigma}$ is the slope.

Germination percentages plotted against storage times yield a sigmoid seed survival curve which is converted to a linear relationship by the probit transformation with slope $\frac{1}{\sigma}$.

The slope is determined as follows:

$$\sigma = 10^{K_E - C_W \log m - C_H t - C_Q t^2}$$

Where, m is the moisture content (fresh weight basis), t is the temperature and K_E, C_W, C_H and C_Q are the species-specific seed viability constants.

On the basis of the the improved seed viability equation, v, p, m and t can be estimated as follows:

$$v = K_i - \frac{p}{\sigma}$$

$$p = \sigma(K_i - v)$$

$$m = 10^{\left[\frac{K_E - C_H t - C_Q t^2 - \log\left(\frac{p}{K_i - v}\right)}{C_W}\right]}$$

$$t = \frac{-C_H \pm \sqrt{C_H^2 - 4C_Q\left(C_W \log m - K_E + \log\left(\frac{p}{K_i - v}\right)\right)}}{2C_Q}$$

The value of the species-specific seed viability constants can be specified either directly in the arguement `vcdirect` or as the index value of the required seed viability constants from the `viabilityconstants` dataset through the argument `vcindex`.

The value of this prediction is appropriate for temperature between -20 to 90 °C and seed moisture content between 5 to 25%. For values beyond this range, a warning will be displayed.

Value

For `FinalViability`, the final viability (%).

For `StorageMC`, the storage moisture content (%).

For `StorageTemp`, the storage temperature (°C).

For `StoragePeriod`, the duration of storage (according to argument `years`).

Note

For initial and/or final viability percentage values of 0% and 100%, adjust it according to sample size using the `PercentAdjust` function to avoid infinity or extreme values in output.

References

Ellis R and Roberts E (1980). "Improved equations for the prediction of seed longevity." *Annals of Botany*, **45**(1), pp. 13–30.

See Also

Sigma, PercentAdjust

Examples

```
# Fetch the index from viabilityconstants dataset
viabilityconstants[grepl("oryza", x = viabilityconstants$Species,
                         ignore.case = TRUE),]

#----------------------------------------------------------------------------
# Final viability
#----------------------------------------------------------------------------
# Use index  87
FinalViability(initial = 98, period = 365, vcindex = 87, mc = 10, temp = 5,
               years = FALSE)
FinalViability(initial = 98, period = 1, vcindex = 87, mc = 10, temp = 5,
               years = TRUE)

# Input the viability constants directly
FinalViability(initial = 98, period = 365,
               vcdirect = c(8.242, 4.345, 0.0307, 0.000501),
               mc = 10, temp = 5, years = FALSE)
FinalViability(initial = 98, period = 1,
               vcdirect = c(8.242, 4.345, 0.0307, 0.000501),
               mc = 10, temp = 5, years = TRUE)

## Not run:
# Error if initial viability is beyond limits (0-100 %)
FinalViability(initial = 110, period = 365, vcindex = 87, mc = 10, temp = 5)

## End(Not run)

# Warning if moisture content is beyond limits (0-100 %)
FinalViability(initial = 98, period = 365, vcindex = 87, mc = 110, temp = 5)

# Warning if temperature is beyond limits (-20 to 90 degree C)
FinalViability(initial = 98, period = 365, vcindex = 87, mc = 10, temp = 95)

# With initial viability 100
FinalViability(initial = 100, period = 365, vcindex = 87, mc = 10, temp = 5,
               years = FALSE)
FinalViability(initial = 100, period = 1, vcindex = 87, mc = 10, temp = 5,
               years = TRUE)

# With intial viability of 100%, use of PercentAdjust() to avoid extremes
FinalViability(initial = PercentAdjust(100, n = 50), period = 365,
               vcindex = 87, mc = 10, temp = 5, years = FALSE)
FinalViability(initial = PercentAdjust(100, n = 50), period = 1,
               vcindex = 87, mc = 10, temp = 5, years = TRUE)

#----------------------------------------------------------------------------
# Storage moisture content
#----------------------------------------------------------------------------
# Use index  87
StorageMC(initial = 98, final = 95, period = 3650, vcindex = 87, temp = 5,
          years = FALSE)
```

```
StorageMC(initial = 98, final = 95, period = 10, vcindex = 87, temp = 5,
          years = TRUE)

# Input the viability constants directly
StorageMC(initial = 98, final = 95, period = 3650,
          vcdirect = c(8.242, 4.345, 0.0307, 0.000501),
          temp = 5, years = FALSE)
StorageMC(initial = 98, final = 95, period = 10,
          vcdirect = c(8.242, 4.345, 0.0307, 0.000501),
          temp = 5, years = TRUE)

## Not run:
# Error if initial viability is beyond limits (0-100 %)
StorageMC(initial = 110, final = 95, period = 3650, vcindex = 87, temp = 5)

# Error if final viability is beyond limits (0-100 %)
StorageMC(initial = 98, final = -10, period = 3650, vcindex = 87, temp = 5)

## End(Not run)

# Warning if temperature is beyond limits (-20 to 90 degree C)
StorageMC(initial = 98, final = 95, period = 3650, vcindex = 87, temp = 95)

#----------------------------------------------------------------------------
# Storage temperature
#----------------------------------------------------------------------------
# Use index  87

# In Celsius
StorageTemp(initial = 98, final = 95, period = 3650, vcindex = 87, mc = 8,
          years = FALSE)
StorageTemp(initial = 98, final = 95, period = 10, vcindex = 87, mc = 8,
          years = TRUE)

# In Fahrenheit
StorageTemp(initial = 98, final = 95, period = 3650, vcindex = 87, mc = 8,
            years = FALSE, unit = "fahrenheit")
StorageTemp(initial = 98, final = 95, period = 10, vcindex = 87, mc = 8,
            years = TRUE, unit = "fahrenheit")

# Input the viability constants directly
StorageTemp(initial = 98, final = 95, period = 3650,
          vcdirect = c(8.242, 4.345, 0.0307, 0.000501),
          mc = 8, years = FALSE)
StorageTemp(initial = 98, final = 95, period = 10,
          vcdirect = c(8.242, 4.345, 0.0307, 0.000501),
          mc = 8, years = TRUE)

## Not run:
# Error if initial viability is beyond limits (0-100 %)
StorageTemp(initial = 110, final = 95, period = 3650, vcindex = 87, mc = 8)

# Error if final viability is beyond limits (0-100 %)
StorageTemp(initial = 98, final = -10, period = 3650, vcindex = 87, mc = 8)

## End(Not run)
```

```
# Warning if moisture content is beyond limits (0-100 %)
StorageTemp(initial = 98, final = 95, period = 3650, vcindex = 87, mc = 110)


#-----------------------------------------------------------------------------
# Storage period
#-----------------------------------------------------------------------------
# Use index  87
 StoragePeriod(initial = 98, final = 95, vcindex = 87, mc = 10, temp = 5,
                years = FALSE)
 StoragePeriod(initial = 98, final = 95, vcindex = 87, mc = 10, temp = 5,
                years = TRUE)

 # Input the viability constants directly
 StoragePeriod(initial = 98, final = 95,
                vcdirect = c(8.242, 4.345, 0.0307, 0.000501),
                mc = 10, temp = 5, years = FALSE)
 StoragePeriod(initial = 98, final = 95,
                vcdirect = c(8.242, 4.345, 0.0307, 0.000501),
                mc = 10, temp = 5, years = TRUE)

 ## Not run:
 # Error if initial viability is beyond limits (0-100 %)
 StoragePeriod(initial = 110, final = 95, vcindex = 87, mc = 10, temp = 5)

 # Error if final viability is beyond limits (0-100 %)
 StoragePeriod(initial = 98, final = -5, vcindex = 87, mc = 10, temp = 5)

## End(Not run)


 # Warning if moisture content is beyond limits (0-100 %)
 StoragePeriod(initial = 98, final = 95, vcindex = 87, mc = 110, temp = 5)

 # Warning if temperature is beyond limits (-20 to 90 degree C)
 StoragePeriod(initial = 98, final = 95, vcindex = 87, mc = 10, temp = 95)

 # With initial viability 100
 StoragePeriod(initial = 100, final = 95, vcindex = 87, mc = 10, temp = 5,
                years = FALSE)
 StoragePeriod(initial = 100, final = 95, vcindex = 87, mc = 10, temp = 5,
                years = TRUE)

 # With intial viability of 100%, use of PercentAdjust() to avoid extremes
 StoragePeriod(initial = PercentAdjust(100, n = 50), final = 95,
                vcindex = 87, mc = 10, temp = 5, years = FALSE)
 StoragePeriod(initial = PercentAdjust(100, n = 50), final = 95,
                vcindex = 87, mc = 10, temp = 5, years = TRUE)
```

Ke *Viability constant K_E based on the species-specific temperature coefficients*

Description

`Ke` computes the viability constant K_E from the species-specific temperature coefficients in case of storage experiments conducted at constant temperature and varying moisture contents.

Usage

```
Ke(K, temp, temp.coeff = c(0.0329, 0.000478))
```

Arguments

`K`	The constant K associated with the relationship of temperature with seed longevity (see **Details**).
`temp`	Temperature in °C.
`temp.coeff`	The species-specific temperature coefficients (C_H and C_Q.) as a numeric vector of length 2.

Details

From seed storage experiments involving storage of seeds at a constant temperature in a range of moisture contents, the effect of moisture content on seed longevity (σ) can be estimated from the following linear relationship:

$$\log \sigma = K - C_w \log m$$

Where, K is the intercept, C_W is the slope and m is the moisture content.

The constant K associated with the relationship of temperature with seed longevity as follows:

$$K = K_E - C_H t - C_Q t^2$$

Where, K_E, C_H and C_Q are the species-specific seed viability constants.

The constant K_E can be estimated from the universal temperature constants (C_H = 0.0329 and C_Q = 0.000478) in case of seed storage experiments at constant temperature and varying moisture content as follows:

$$K_E = K + C_H t + C_Q t^2$$

Value

The value of species-specific seed viability constant K_E.

References

Ellis R and Roberts E (1980). “Improved equations for the prediction of seed longevity.” *Annals of Botany*, **45**(1), pp. 13–30. Pritchard HW and Dickie JB (2003). “Predicting seed longevity: The use and abuse of seed viability equations.” In Smith RD, Dickie JB, Linington SH, Pritchard HW and Probert RJ (eds.), *Seed Conservation: Turning Science into Practice*, pp. 653–721. Kew, UK, Royal Botanic Gardens.

See Also

`Sigma`

Examples

```
Ke(36, 10)
```

P50 *Half-viability period*

Description

`P50` computes the half-viability period, which is the time taken for 50% of the seeds to lose viability.

Usage

```
P50(initial, vcindex, vcdirect, mc, temp, years = FALSE)
```

Arguments

`initial`	The initial viability (%).
`vcindex`	An interger value indicating the index of seed viability. constants to be used from the `viabilityconstants` dataset in the package.
`vcdirect`	A numeric vector of length 4 with the four viability constants *viz.*: K_E, C_W, C_H and C_Q.
`mc`	Moisture content.
`temp`	Temperature in °C.
`years`	If `TRUE`, returns the output period in years instead of days.

Details

The period to lose 50% viability (P_{50}) is computed according to the relationship between probit percentage viabilities and time of storage described by Ellis and Roberts (1980) as follows:

$$v = K_i - \frac{p}{\sigma}$$

or

$$v = K_i - \left(\frac{1}{\sigma}\right) \cdot p$$

Where, v is the probit percentage viability at storage time p (final viability), K_i is the probit percentage viability of the seedlot at the beginning of storage (seedlot constant) and $\frac{1}{\sigma}$ is the slope.

Germination percentages plotted against storage times yield a sigmoid seed survival curve which is converted to a linear relationship by the probit transformation with slope $\frac{1}{\sigma}$.

When v = 0 (equivalent to 50% viability), P_{50} can be computed as follows:

$$P_{50} = K_i \times \sigma$$

If the initial viablity (`initial`) is beyond limits (0-100 %, an error is issued.

The value of this computation is appropriate for temperature between -20 to 90 °C and seed moisture content between 5 to 25%. For values beyond this range, a warning will be displayed.

Value

The half-viability period in days or years (according to argument `years`).

Note

For initial viability percentage values of 100%, adjust it according to sample size using the `PercentAdjust` function to avoid infinity values in output.

References

Ellis R and Roberts E (1980). “Improved equations for the prediction of seed longevity.” *Annals of Botany*, **45**(1), pp. 13–30.

See Also

`Sigma`, `PercentAdjust`

Examples

```
P50(initial = 98, vcindex = 24, mc = 5, temp = -20)
P50(initial = 98, vcindex = 24, mc = 5, temp = -20, years = TRUE)

# With intial viability of 100%
P50(initial = 100, vcindex = 24, mc = 5, temp = -20)
P50(initial = 100, vcindex = 24, mc = 5, temp = -20, years = TRUE)

# With intial viability of 100%, use of PercentAdjust() to avoid Inf
P50(initial = PercentAdjust(100, n = 50), vcindex = 24, mc = 5, temp = -20)
P50(initial = PercentAdjust(100, n = 50), vcindex = 24, mc = 5, temp = -20,
    years = TRUE)

## Not run:
# Error if initial viability is beyond limits (0-100 %)
P50(initial = 110, vcindex = 24, mc = 5, temp = -20)

## End(Not run)
```

`Percent2Probit` *Probit transformation*

Description

These functions transfrom data between percentage, probit and Normal Equivalent Deviate (NED).

Usage

```
PercentAdjust(percentage, n)

Percent2NED(percentage)

Percent2Probit(percentage)
```

```
Probit2NED(probit)

NED2Probit(NED)

NED2Percent(NED)

Probit2Percent(probit)
```

Arguments

`percentage`	The percentage value.
`n`	Sample size for estimation of percentage.
`probit`	The probit value
`NED`	The NED value.

Details

Probit transformation can be used to transform a sigmoid curve of percentage data to a linear one. The probit transformation is defined as $NED + 5$. However the two terms probit and NED are used interchangeably in literature.

NED function (Φ^{-1}) is is the inverse of the cumulative distribution function (Φ) of the standard normal distribution ($z \sim N(0, 1)$) or the quantile function associated with the standard normal distribution.

For percentage p,

$$\mathrm{NED}(p) = \Phi^{-1}(p) = \sqrt{2}\,\mathrm{erf}^{-1}(2p - 1)$$

and

$$\mathrm{probit}(p) = \mathrm{NED}(p) + 5$$

The `PercentAdjust` function adjusts the percentage values of 0 and 100 to $100 \times \frac{0.25}{n}$ and $100 \times \frac{n-0.25}{n}$ respectively, according to the sample size n to avoid infinity values during probit transformation (Miller and Tainter, 1944).

Value

The transformed value.

References

Bliss CI (1934). "The method of probits." *Science*, **79**(2037), pp. 38–39. doi: 10.1126/science.79.2037.38.

Miller LC and Tainter ML (1944). "Estimation of the ED50 and its error by means of logarithmic-probit graph paper." *Proceedings of the Society for Experimental Biology and Medicine*, **57**(2), pp. 261–264. doi: 10.3181/003797275714776.

Finney DJ (ed.) (1952). *Probit Analysis: A Statistical Treatment of the Sigmoid Response Curve*. Cambridge University Press, Cambridge, England.

Examples

```
Percent2NED(0:100)
Percent2Probit(0:100)

Percent2NED(25)
Percent2Probit(25)
Percent2NED(25) +5
NED2Probit(-0.6744898)

# Percentage adjustment for 0 and 100
Percent2Probit(100)
Percent2Probit(0)
n = 50
Percent2Probit(PercentAdjust(100, n))
Percent2Probit(PercentAdjust(0, n))
```

SeedEqMC *Seed equilibrium moisture content and equilibrium relative humidity of the seed storage environment*

Description

Compute the following metrics:

- `SeedEqMC` The seed equilibrium moisture content from known environmental conditions and oil content.
- `EqRH` The equilibrium relative humidity of the seed storage environment from seed equilibrium moisture content, oil content and temperature.

Usage

```
SeedEqMC(oilcontent, rh, temp, basis = c("wet", "dry"))

EqRH(oilcontent, mc, temp, basis = c("wet", "dry"))
```

Arguments

`oilcontent`	The percentage oil content of seed (dry basis).
`rh`	Relative humidity expressed in percentage.
`temp`	Temperature in °C.
`basis`	The type of estimation of moisture content specified in the argument `mc`. Either `"wet"` or `"dry"`.
`mc`	The seed equilibrium moisture content on wet or dry basis (according to argument `basis`).

Details

This relationship between seed equilibrium moisture content, seed oil content, the equilibrium relative humidity and temperature of the storage environment was described by Cromarty et al., (1982) as follows:

$$(1-R) = \mathrm{e}^{-\left(\frac{\left[\frac{M_e \times (1.1+\frac{T}{90})}{1-D_O}\right]^2}{440}\right)}$$

Where, R is the relative humidity expressed as decimal, M_e is the equilibrium percentage moisture content (dry basis), T is the temperature in °C of air or the seed equilibrium, D_O is the oil content of seed (dry basis) expressed as decimal and e is the mathematical constant 2.718282.

For values of oil content (`oilcontent`), relative humidity (`rh`) and seed equilibrium moisture content (`mc`) beyond the limits of 0-100 %, a warning is issued.

Value

For `SeedEqMC`, the seed equilibrium moisture content on wet or dry basis (according to argument `basis`) expressed in percentage.

For `EqRH`, the equilibrium relative humidity expressed in percentage.

Note

The above expression by Cromarty et al., (1982) is recommended for temperature and humidity ranges of 0-40 °C and 10-70% RH for starchy seeds (eg. cereals); and 15-25 °C and 10-70% RH for oilseeds.

References

Cromarty A, Ellis RH and Roberts EH (1982). *The Design of Seed Storage Facilities for Genetic Conservation*, Revised 1985 and 1990 edition. International Board for Plant Genetic Resources, Rome, Italy.

See Also

`wet2dry`

Examples

```
SeedEqMC(oilcontent = 29, rh = 13, temp = 25, basis = "wet")
SeedEqMC(oilcontent = 29, rh = 13, temp = 25, basis = "dry")

EqRH(oilcontent = 29, mc = 5, temp = 25, basis = "wet")
EqRH(oilcontent = 29, mc = 5, temp = 25, basis = "dry")

# Warning if oilcontent is beyond limits (0-100 %)
SeedEqMC(oilcontent = 125, rh = 13, temp = 25, basis = "wet")
EqRH(oilcontent = 125, mc = 5, temp = 25, basis = "wet")

# Warning if relative humidity is beyond limits (0-100 %)
SeedEqMC(oilcontent = 29, rh = 115, temp = 25, basis = "wet")

# Warning if moisture content is beyond limits (0-100 %)
```

```
EqRH(oilcontent = 29, mc = 115, temp = 25, basis = "wet")
```

Sigma *Period to lose unit probit viability*

Description

`Sigma` calculates the period to lose one probit viability (σ) under storage at a given moisture content and temperature.

Usage

```
Sigma(vcindex, vcdirect, mc, temp, years = FALSE)
```

Arguments

`vcindex`	An interger value indicating the index of seed viability. constants to be used from the `viabilityconstants` dataset in the package.
`vcdirect`	A numeric vector of length 4 with the four viability constants *viz.*: K_E, C_W, C_H and C_Q.
`mc`	Moisture content.
`temp`	Temperature in °C.
`years`	If `TRUE`, returns the output period in years instead of days.

Details

This function computes the period to lose one probit viability (σ) according to the improved seed viability equation of Ellis and Roberts (1980) as follows:

$$v = K_i - \frac{p}{\sigma}$$

or

$$v = K_i - \left(\frac{1}{\sigma}\right) \cdot p$$

Where, v is the probit percentage viability at storage time p (final viability), K_i is the probit percentage viability of the seedlot at the beginning of storage (seedlot constant) and $\frac{1}{\sigma}$ is the slope.

Germination percentages plotted against storage times yield a sigmoid seed survival curve which is converted to a linear relationship by the probit transformation with slope $\frac{1}{\sigma}$.

The slope is determined as follows:

$$\sigma = 10^{K_E - C_W \log m - C_H t - C_Q t^2}$$

Where, v is the probit percentage viability at storage time p (final viability), K_i is the probit percentage viability of the seedlot at the beginning of storage (seedlot constant), m is the moisture content (fresh weight basis), t is the temperature and K_E, C_W, C_H and C_Q are the species-specific seed viability constants.

The value of the species-specific seed viability constants can be specified either directly in the arguement vcdirect or as the index value of the required seed viability constants from the viabilityconstants dataset through the argument vcindex.

The value of this prediction is appropriate for temperature between -20 to 90 °C and seed moisture content between 5 to 25%. For values beyond this range, a warning will be displayed.

Value

The period to lose one probit in days or years (according to argument years).

References

Ellis R and Roberts E (1980). "Improved equations for the prediction of seed longevity." *Annals of Botany*, **45**(1), pp. 13–30.

Examples

```
#---------------------------------------------------------------------------
# Days/Years to lose unit probit viability for rice seeds stored at
# 5 degree celsius and 10% moisture content.
#---------------------------------------------------------------------------
# Fetch the index from viabilityconstants dataset
viabilityconstants[grepl("oryza", x = viabilityconstants$Species,
                         ignore.case = TRUE),]
# Use index 87
Sigma(vcindex = 87, mc = 10, temp = 5)
Sigma(vcindex = 87, mc = 10, temp = 5, years = TRUE)

# Input the viability constants directly
Sigma(vcdirect = c(8.242, 4.345, 0.0307, 0.000501), mc = 10, temp = 5)
Sigma(vcdirect = c(8.242, 4.345, 0.0307, 0.000501), mc = 10, temp = 5,
      years = TRUE)

# Warning if moisture content is beyond limits (0-100 %)
Sigma(vcindex = 87, mc = 110, temp = 5)

# Warning if temperature is beyond limits (-20 to 90 degree C)
Sigma(vcindex = 87, mc = 10, temp = 95)

#---------------------------------------------------------------------------
# Days/Years to lose unit probit viability for soybean seeds stored at
# -18 degree celsius and 8% moisture content.
#---------------------------------------------------------------------------
# Fetch the index from viabilityconstants dataset
viabilityconstants[grepl("glycine", x = viabilityconstants$Species,
                         ignore.case = TRUE),]
# Use index  59
Sigma(vcindex = 59, mc = 8, temp = -18)
Sigma(vcindex = 59, mc = 8, temp = -18, years = TRUE)

# Input the viability constants directly
Sigma(vcdirect = c(7.292, 3.996, 0.0295, 0.000491), mc = 8, temp = -18)
Sigma(vcdirect = c(7.292, 3.996, 0.0295, 0.000491), mc = 8, temp = -18,
      years = TRUE)

# Warning if moisture content is beyond limits (0-100 %)
```

```
Sigma(vcindex = 59, mc = 110, temp = 5)

# Warning if temperature is beyond limits (-20 to 90 degree C)
Sigma(vcindex = 59, mc = 10, temp = 95)
```

`SigmaTransformed` *SigmaTransformed*

Description

`SigmaTransformed` transforms the measured sigma value at a specific temperature to an estimate of sigma at another temperature. This useful in comparitive seed testing protocol to compare seed longevities among species tested at different temperatures (Probert et al., 2009).

Usage

```
SigmaTransformed(sigma, temp1, temp2, temp.coeff = c(0.0329, 0.000478))
```

Arguments

`sigma` The inverse of slope from the seed viability equation (σ), estimated at temperature `temp1`.

`temp1` The temperature at which `sigma` is estimated in °C.

`temp2` The temperature at which the transformed σ is to be estimated in °C.

`temp.coeff` The species-specific temperature coefficients (C_H and C_Q.) as a numeric vector of length 2.

Details

The transformation is based on the effect of temperature on seed longevity (σ) (identified by storage experiment with constant moisture content and varying temperature) which is as follows:

$$\log \sigma = \beta - C_H t - C_H t^2$$

Where, C_H and C_Q are the species-specific temperature coefficients, t is the temperature and β is the constant associated with moisture relations of seed longevity.

Value

The transformed value of σ at temperature `temp2`.

References

Ellis R and Roberts E (1980). "Improved equations for the prediction of seed longevity." *Annals of Botany*, **45**(1), pp. 13–30. Probert RJ, Daws MI and Hay FR (2009). "Ecological correlates of ex situ seed longevity: A comparative study on 195 species." *Annals of Botany*, **104**(1), pp. 57–69. doi: 10.1093/aob/mcp082, `https://academic.oup.com/aob/article/104/1/57/122605`.

See Also

`Sigma`

Examples

```
SigmaTransformed(250, 60, 45)
```

`viabilityconstants` *Viability constants.*

Description

A dataset of the known species-specific seed viability constants estimated from the 'improved seed viability equation' of Ellis and Roberts (1980).

Usage

```
viabilityconstants
```

Format

A data frame with 7 columns:

Index A unique index value, called by the `vcindex` argument in different functions.

Species The species for which the constants were estimated.

Ke The species-specific moisture constant K_E

Cw The species-specific moisture constant C_W

Ch The species-specific temperature constant C_H

Cq The species-specific temperature constant C_Q

Reference Source of the data.

Details

Various functions such as `Sigma`, `StorageMC` use the viability constants from this dataset for computation through the argument `vcindex`.

Note

Viability constants derived from storage experiment at a single temperature are highlighted by "*" in species column. In such cases, C_H and C_Q are the universal temperature coefficients (0.0329 and 0.000478 respectively).

Source

Primarily from Hong et al., 1996, updated with more species. The source for each record is mentioned in the **Reference** column.

References

Ellis R and Roberts E (1980). "Improved equations for the prediction of seed longevity." *Annals of Botany*, **45**(1), pp. 13–30.

Hong TD, Linington S and Ellis RH (1996). *Seed Storage Behaviour: A Compendium*, number 4 series Handbooks for Genebanks. International Plant Genetic Resources Institute (IPGRI), Rome, Italy.

Examples

```
# Soybean
viabilityconstants[grepl("glycine", x = viabilityconstants$Species,
                         ignore.case = TRUE),]
# Rice
viabilityconstants[grepl("oryza", x = viabilityconstants$Species,
                         ignore.case = TRUE),]
# Wheat
viabilityconstants[grepl("triticum", x = viabilityconstants$Species,
                         ignore.case = TRUE),]
## Not run:
# View dataset
View(viabilityconstants)

## End(Not run)
```

Index