Package ‘viabilitymetrics’

January 31, 2019

Type Package

Title Seed Viability Calculations and Curve Fitting

Version 0.0.0.9100

Description An implementation of viability equations of Ellis and Roberts (1980) <doi:10/gcshwj> and Mead and Grey (1999) <doi:10/gcsgt7> for seed viability curve fitting and 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 (>= 3.5.0)

VignetteBuilder knitr

RoxygenNote 6.1.1

Imports broom,
data.table,
dplyr,
ggplot2,
gnm,
plyr,
Rdpack

Suggests knitr,
rmarkdown,
pander

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 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 AS, Ellis RH, 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 integer 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 RH, Roberts EH (1980). "Improved equations for the prediction of seed longevity." *Annals of Botany*, **45**(1), 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)
```

FitSigma *Seed viability curve fitting to estimate K_i and σ*

Description

Fit seed viability/survival curve to estimate the seed lot constant (K_i) and the period to lose unit probit viability (σ).

Usage

```
FitSigma(data, viability.percent, samp.size, storage.period,
  probit.method = c("glm", "tflm"), use.cv = FALSE,
  control.viability = 100)
```

Arguments

`data` A data frame with the seed viability data recorded periodically. It should possess columns with data on

- Viability percentage (to be indicated by the argument `viability.percent`),
- Sample size (to be indicated by the argument `samp.size`) and
- Storage period (to be indicated by the argument `storage.period`).

`viability.percent`
The name of the column in `data` with the viability percentages as a character string.

`samp.size` The name of the column in `data` with the sample size used for calculating viability percentages as a character string.

`storage.period` The name of the column in `data` with the time periods at which the viabilty percentages was recorded as a character string.

`probit.method` The method to be used for fitting seed viability curve. Either as a generalised linear model with a probit link function (`"glm"`, recommended) or as a linear model with probit transformed viability percentages (`"tflm"`).

`use.cv` logical. If `TRUE`, then the percentage value specified in the `control.viabilty` argument is incorporated as the control viability parameter into the seed viability equation for fitting. Default is `FALSE`.

`control.viability`
The control viability (%).

Details

This function fits seed survival data to the following seed viability equation (Ellis and Roberts 1980) which models the relationship between probit percentage viability and time period of storage.

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

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.

The above equation may be expressed as a generalized linear model (GLM) with a probit (cumulative normal distribution) link function as follows (Hay et al. 2014).

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

Where, y is the proportion of seeds viabile after time period p and the link function is ϕ^{-1}, the inverse of the cumulative normal distribution function.

The parameters estimated are the intercept K_i, theoretical viability of the seeds at the start of storage or the seed lot constant, and the slope $-\sigma^{-1}$, where σ is the standard deviation of the normal distribution of seed deaths in time or the period of time to lose unit probit viability.

This function can also incorporate a control viability parameter into the model to fit the modified model suggested by (Mead and Gray 1999). The modified model is as follows.

$$y = C_v \times \phi(v) = C_v \times \phi\left(K_i - \left(\frac{1}{\sigma}\right)p\right)$$

Where, C_v is the control viability parameter which is the proportion of respondent seeds. This excludes the bias due to seeds of the ageing population that have already lost viability at the start of storage and those non-respondent seeds that are not part of the ageing population due to several reasons.

Value

A list of class `FitSigma` with the following components:

`data`	A data frame with the data used for computing the model.
`model`	The fitted model as an object of class `glm` (if `probit.method = "glm"`) or `lm` (if `probit.method = "lm"`).
`parameters`	A data.frame of parameter estimates, standard errors and p value.
`fit`	A one-row data frame with estimates of model fitness such as log likelyhoods, Akaike Information Criterion, Bayesian Information Criterion, deviance and residual degrees of freedom.
`Ki`	The estimated seed lot constant from the model.
`sigma`	The estimated period of time to lose unit probit viability from the model.
`message`	Warning or error messages generated during fitting of model, if any.

References

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

Hay FR, Mead A, Bloomberg M (2014). "Modelling seed germination in response to continuous variables: use and limitations of probit analysis and alternative approaches." *Seed Science Research*, **24**(3), 165–186.

Mead A, Gray D (1999). "Prediction of seed longevity: A modification of the shape of the Ellis and Roberts seed survival curves." *Seed Science Research*, **9**(1), 63–73.

Examples

```
data(seedsurvival)
df <- seedsurvival[seedsurvival$crop == "Groundnut" &
                     seedsurvival$mc == 7 &
                     seedsurvival$temp == 25,
                   c("period", "rep", "viabilitypercent", "sampsize")]

plot(df$period, df$viabilitypercent)

#----------------------------------------------------------------------------
# Generalised linear model with probit link function (without cv)
#----------------------------------------------------------------------------
model1a <- FitSigma(data = df, viability.percent = "viabilitypercent",
                   samp.size = "sampsize", storage.period = "period",
                   probit.method = "glm")
model1a
# Raw model
model1a$model

# Model parameters
model1a$parameters

# Model fit
model1a$fit

#----------------------------------------------------------------------------
# Generalised linear model with probit link function (with cv)
#----------------------------------------------------------------------------
model1b <- FitSigma(data = df, viability.percent = "viabilitypercent",
                   samp.size = "sampsize", storage.period = "period",
                   probit.method = "glm",
                   use.cv = TRUE, control.viability = 98)
model1b
# Raw model
model1b$model

# Model parameters
model1b$parameters

# Model fit
model1b$fit

#----------------------------------------------------------------------------
```

```
# Linear model after probit transformation (without cv)
#----------------------------------------------------------------------------
model2a <- FitSigma(data = df, viability.percent = "viabilitypercent",
                   samp.size = "sampsize", storage.period = "period",
                   probit.method = "tflm")
model2a
# Raw model
model2a$model

# Model parameters
model2a$parameters

# Model fit
model2a$fit

#----------------------------------------------------------------------------
# Linear model after probit transformation (with cv)
#----------------------------------------------------------------------------
model2b <- FitSigma(data = df, viability.percent = "viabilitypercent",
                   samp.size = "sampsize", storage.period = "period",
                   probit.method = "tflm",
                   use.cv = TRUE, control.viability = 98)
model2b
# Raw model
model2b$model

# Model parameters
model2b$parameters

# Model fit
model2b$fit
```

`FitSigma.batch` *Seed viability curve fitting to estimate multiple values of K_i and σ according to a grouping variable*

Description

Fit seed viability/survival curve to estimate multiple values of the seed lot constant (K_i) and the period to lose unit probit viability (σ) according to a grouping variable.

Usage

```
FitSigma.batch(data, group, ...)
```

Arguments

`data` A data frame with the seed viability data recorded periodically. It should possess columns with data on

- Viability percentage (to be indicated by the argument `viability.percent`),
- Sample size (to be indicated by the argument `samp.size`),
- Storage period (to be indicated by the argument `storage.period`) and

- Grouping variable (to be indicated by the argument group).

group
: The name of the column in data with grouping variable as a character string.

...
: Arguments to be passed on to FitSigma.

Value

A list of class FitSigma.batch with the following components:

data
: A data frame with the data used for computing the models.

models
: A data frame with the group-wise values of model parameters, K_i and σ and the fit statistics.

See Also

FitSigma

Examples

```
data(seedsurvival)
df <- seedsurvival[seedsurvival$mc == 7 & seedsurvival$temp == 25,
                   c("crop", "period", "rep",
                     "viabilitypercent", "sampsize")]

#----------------------------------------------------------------------------
# Generalised linear model with probit link function (without cv)
#----------------------------------------------------------------------------
model1a <- FitSigma.batch(data = df, group = "crop",
                          viability.percent = "viabilitypercent",
                          samp.size = "sampsize", storage.period = "period",
                          probit.method = "glm")
model1a

#----------------------------------------------------------------------------
# Generalised linear model with probit link function (with cv)
#----------------------------------------------------------------------------
model1b <- FitSigma.batch(data = df, group = "crop",
                          viability.percent = "viabilitypercent",
                          samp.size = "sampsize", storage.period = "period",
                          probit.method = "glm",
                          use.cv = TRUE, control.viability = 98)
model1b

#----------------------------------------------------------------------------
# Linear model after probit transformation (without cv)
#----------------------------------------------------------------------------
model2a <- FitSigma.batch(data = df, group = "crop",
                          viability.percent = "viabilitypercent",
                          samp.size = "sampsize", storage.period = "period",
                          probit.method = "tflm")
model2a

#----------------------------------------------------------------------------
# Linear model after probit transformation (with cv)
#----------------------------------------------------------------------------
model2b <- FitSigma.batch(data = df, group = "crop",
```

```
                          viability.percent = "viabilitypercent",
                          samp.size = "sampsize", storage.period = "period",
                          probit.method = "tflm",
                          use.cv = TRUE, control.viability = 98)
model2b
```

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 RH, Roberts EH (1980). "Improved equations for the prediction of seed longevity." *Annals of Botany*, **45**(1), 13–30.

Pritchard HW, Dickie JB (2003). "Predicting seed longevity: The use and abuse of seed viability equations." In Smith RD, Dickie JB, Linington SH, Pritchard HW, Probert RJ (eds.), *Seed Conservation: Turning Science into Practice*, 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 integer 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 RH, Roberts EH (1980). "Improved equations for the prediction of seed longevity." *Annals of Botany*, **45**(1), 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 transform data between percentage, probit and Normal Equivalent Deviate (NED) (Bliss 1934; Finney 1952).

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), 38–39.

Miller LC, 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), 261–264.

Finney DJ (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))
```

`plot.FitSigma` *Plot the fitted seed viability curve from a* `FitSigma` *object*

Description

`plot.FitSigma` plots the fitted seed viability/survival curve from a `FitSigma` object as an object of class `ggplot`.

Usage

```
## S3 method for class 'FitSigma'
plot(x, limits = TRUE, annotate = TRUE, ...)
```

Arguments

`x`	An object of class `FitSigma` obtained as output from the `FitSigma` function.
`limits`	logical. If `TRUE`, set the limits of y axis (viability percentage) between 0 and 100 in the viability curve plot. If `FALSE`, limits are set according to the data. Default is `TRUE`.
`annotate`	logical. If `TRUE`, K_i and σ values are annotated on the plot. Default is `TRUE`.
`...`	Default plot arguments.

Value

The plot of the seed viability curve as an object of class `ggplot`.

See Also

FitSigma

Examples

```
data(seedsurvival)
df <- seedsurvival[seedsurvival$crop == "Groundnut" &
                     seedsurvival$mc == 7 &
                     seedsurvival$temp == 25,
                   c("period", "rep", "viabilitypercent", "sampsize")]

plot(df$period, df$viabilitypercent)

#----------------------------------------------------------------------------
# Generalised linear model with probit link function (without cv)
#----------------------------------------------------------------------------
model1a <- FitSigma(data = df, viability.percent = "viabilitypercent",
                   samp.size = "sampsize", storage.period = "period",
                   probit.method = "glm")
plot(model1a)

#----------------------------------------------------------------------------
# Generalised linear model with probit link function (with cv)
#----------------------------------------------------------------------------
model1b <- FitSigma(data = df, viability.percent = "viabilitypercent",
                   samp.size = "sampsize", storage.period = "period",
                   probit.method = "glm",
                   use.cv = TRUE, control.viability = 98)
plot(model1b)

#----------------------------------------------------------------------------
# Linear model after probit transformation (without cv)
#----------------------------------------------------------------------------
model2a <- FitSigma(data = df, viability.percent = "viabilitypercent",
                   samp.size = "sampsize", storage.period = "period",
                   probit.method = "tflm")
plot(model2a)

#----------------------------------------------------------------------------
# Linear model after probit transformation (with cv)
#----------------------------------------------------------------------------
model2b <- FitSigma(data = df, viability.percent = "viabilitypercent",
                   samp.size = "sampsize", storage.period = "period",
                   probit.method = "tflm",
                   use.cv = TRUE, control.viability = 98)
plot(model2b)
```

plot.FitSigma.batch *Plot the fitted seed viability curves from a* FitSigma.batch *object*

Description

plot.FitSigma.batch plots the group-wise fitted seed viability/survival curves from a FitSigma.batch object as an object of class ggplot.

Usage

```
## S3 method for class 'FitSigma.batch'
plot(x, limits = TRUE, grid = FALSE, ...)
```

Arguments

x	An object of class FitSigma.batch obtained as output from the FitSigma.batch function.
limits	logical. If TRUE, set the limits of y axis (viability percentage) between 0 and 100 in the viability curve plot. If FALSE, limits are set according to the data. Default is TRUE.
grid	logical. If TRUE, a symmetric matrix grid of plots is produced instead of a single plot with multiple curves. Default is FALSE.
...	Default plot arguments.

Value

The plot of the seed viability curves as an object of class ggplot.

See Also

FitSigma.batch

Examples

```
df1 <- data.frame(period = 1:15)
df1$sampsize <- 50

df3 <- df2 <- df1

df1$crop <- "Wheat"
df1$vcindex <- 136
df2$crop <- "Soybean"
df2$vcindex <- 59
df3$crop <- "Tomato"
df3$vcindex <- 78

df <- rbind(df1, df2, df3)
rm(df1, df2, df3)

df$viabilitypercent <- mapply(FinalViability, period = df$period,
                              vcindex = df$vcindex,
                              MoreArgs = list(initial = PercentAdjust(100, 50),
```

```
                                          mc = 7, temp = 25,
                                          years = TRUE),
                              SIMPLIFY = TRUE)

df$viabilitypercent <- round(df$viabilitypercent)
df$viabilitypercent[df$viabilitypercent == 99] <- 100
df$crop <- as.factor(df$crop)

df <- df[, c("crop", "period", "viabilitypercent", "sampsize")]
df[df$crop == "Wheat",]$viabilitypercent <- df[df$crop == "Wheat",]$viabilitypercent - 2


plot(df$period, df$viabilitypercent, col = df$crop)
legend(10, 60, legend=levels(df$crop),
       col = c("black", "red", "green"), pch = 1)

#----------------------------------------------------------------------------
# Generalised linear model with probit link function (without cv)
#----------------------------------------------------------------------------
model1a <- FitSigma.batch(data = df, group = "crop",
                          viability.percent = "viabilitypercent",
                          samp.size = "sampsize", storage.period = "period",
                          probit.method = "glm")
plot(model1a)
plot(model1a, grid = TRUE)

#----------------------------------------------------------------------------
# Generalised linear model with probit link function (with cv)
#----------------------------------------------------------------------------
model1b <- FitSigma.batch(data = df, group = "crop",
                          viability.percent = "viabilitypercent",
                          samp.size = "sampsize", storage.period = "period",
                          probit.method = "glm",
                          use.cv = TRUE, control.viability = 98)
plot(model1b)
plot(model1b, grid = TRUE)

#----------------------------------------------------------------------------
# Linear model after probit transformation (without cv)
#----------------------------------------------------------------------------
model2a <- FitSigma.batch(data = df, group = "crop",
                          viability.percent = "viabilitypercent",
                          samp.size = "sampsize", storage.period = "period",
                          probit.method = "tflm")
plot(model2a)
plot(model2a, grid = TRUE)

#----------------------------------------------------------------------------
# Linear model after probit transformation (with cv)
#----------------------------------------------------------------------------
model2b <- FitSigma.batch(data = df, group = "crop",
                          viability.percent = "viabilitypercent",
                          samp.size = "sampsize", storage.period = "period",
                          probit.method = "tflm",
                          use.cv = TRUE, control.viability = 98)
plot(model2b)
plot(model2b, grid = TRUE)
```

print.FitSigma — *Prints K_i and σ from a* FitSigma *object*

Description

print.FitSigma prints to console the seed lot constant (K_i) and the period to lose unit probit viability (σ).

Usage

```
## S3 method for class 'FitSigma'
print(x, ...)
```

Arguments

x	An object of class print.FitSigma.
...	Unused

Value

The K_i and σ values (degree Celsius) in the console.

See Also

FitSigma

print.FitSigma.batch — *Prints K_i and σ from a* FitSigma.batch *object*

Description

print.FitSigma.batch prints to console the seed lot constant (K_i) and the period to lose unit probit viability (σ).

Usage

```
## S3 method for class 'FitSigma.batch'
print(x, ...)
```

Arguments

x	An object of class print.FitSigma.batch.
...	Unused

Value

The K_i and σ values (degree Celsius) in the console.

See Also

FitSigma.batch

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 AS, Ellis RH, 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")
```

---

seedsurvival *Seed survival dataset*

Description

Seed survial data of three crops - groundnut, soybean and wheat stored for 15 years at combinations of three different moisture contents (3%, 5% and 7%) and temperatures (0 °C, 10 °C and 25 °C).

Usage

```
seedsurvival
```

Format

A data frame with 7 columns:

crop The crop name.

mc The moisture content (%).

temp The temperature (°C).

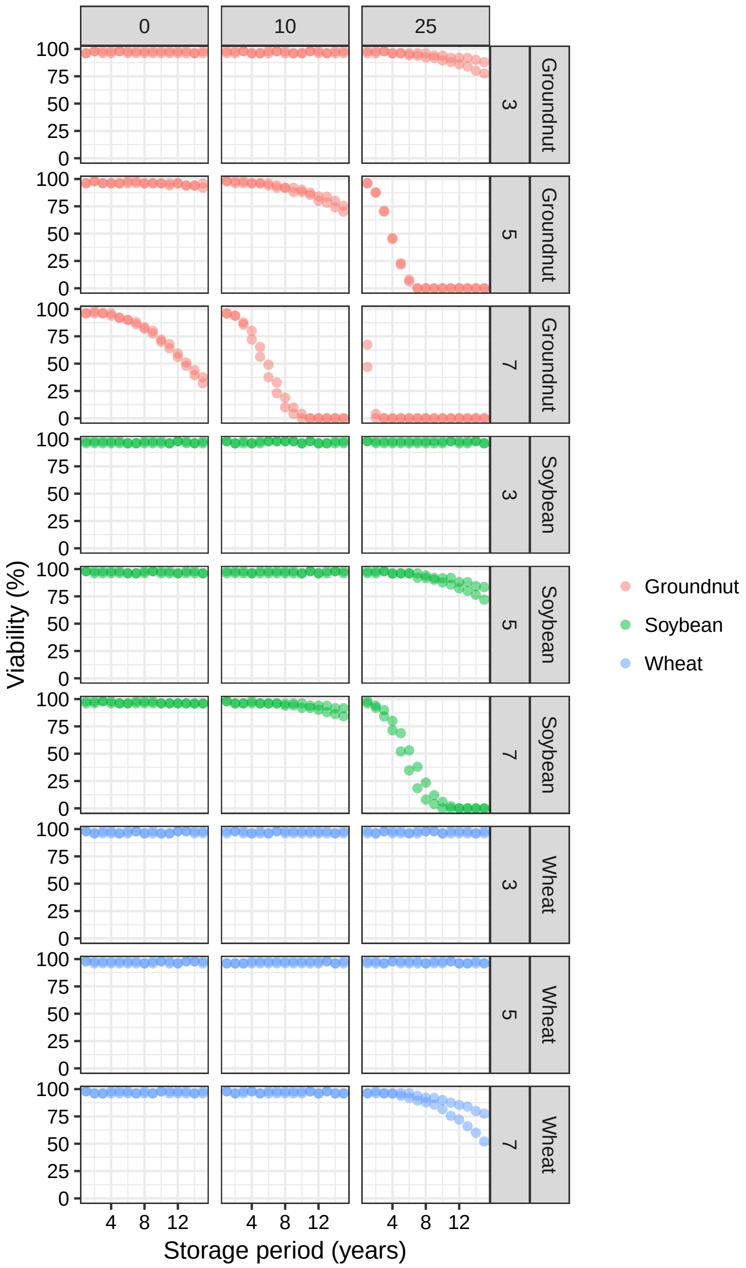
period The year at which viability was recorded.

rep The replication.

viabilitypercent The viability percentage value recorded.

sampsize The sample size used for estimating viability percentage.

Details



Examples

```
library(ggplot2, quietly = TRUE)

ggplot(data = seedsurvival, aes(x = period, y = viabilitypercent)) +
  geom_point(aes(colour = crop), alpha = 0.5) +
  labs(x = "Storage period (years)", y = "Viability (%)") +
  facet_grid(crop + mc ~ temp) +
  theme_bw() +
  theme(axis.text = element_text(colour = "black"),
        legend.title = element_blank())
```

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 integer 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 argument `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 RH, Roberts EH (1980). "Improved equations for the prediction of seed longevity." *Annals of Botany*, **45**(1), 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 RH, Roberts EH (1980). “Improved equations for the prediction of seed longevity.” *Annals of Botany*, **45**(1), 13–30.

Probert RJ, Daws MI, Hay FR (2009). “Ecological correlates of ex situ seed longevity: A comparative study on 195 species.” *Annals of Botany*, **104**(1), 57–69.

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

Ali TS (2014). "Determination of seed viability constants in sorghum under various storage conditions." *Iranian Journal of Field Crop Science (Iranian Journal of Agricultural Sciences)*, **45**(3), 377–387.

Alivand R, Tavakol Afshari R, Sharifzade F (2013). "Germination response and estimation of seed deterioration of *Brassica napus* under various storage conditions." *Iranian Journal of Field Crop Science*, **44**(1), 69–81.

Baladi S, Balouchi H (2016). "Evaluation of *Lallemantia royleana* seed longevity under varying conditions of temperature and moisture content." *Seed Science and Technology*, **44**(2), 320–326.

Balouchi H, Baladi S, Moradi A, Dehnavi MM (2017). "The influence of temperature and moisture content on seed longevity of two genotypes of *Linum usitatissimum*." *Seed Science and Technology*, **45**(1), 130–138.

Bam RK, Hong TD, Ellis RH, Kumaga FK, Asiedu EA (2008). "Storage behaviour of two contrasting upland rice genotypes." *Ghana Journal of Agricultural Science*, **41**(1).

Belletti P, Lanteri S, Lotito S (1991). "The influence of temperature and moisture on seed ageing in Iceland poppy (*Papaver nudicaule* L.)." *Scientia Horticulturae*, **48**(1-2), 153–158.

Bonner FT (1994). "Predicting seed longevity for four forest tree species with orthodox seeds." *Proceedings of the International Seed Testing Association*.

Chaves MMF, Usberti R (2004). "Controlled seed deterioration in *Dalbergia nigra* and *Dimorphandra mollis*, endangered Brazilian forest species." *Seed Science and Technology*, **32**(3), 813–823.

Crawford AD, Hay FR, Plummer JA, Probert RJ, Steadman KJ (2013). "One-step fitting of seed viability constants for two Australian plant species, *Eucalyptus erythrocorys* (Myrtaceae) and *Xanthorrhoea preissii* (Xanthorrhoeacea)." *Australian Journal of Botany*, **61**(1), 1–10.

Daniel IO, Kruse M, Borner A (2011). "Comparative longevity and viability modeling of *Solanum macrocarpon* L. seeds." *Seed Science and Technology*, **39**(3), 680–685.

Daniel IO, Kruse M, Borner A (2013). "Controlled deterioration and predicting viability of okra seed in storage." *International Journal of Vegetable Science*, **19**(4), 324–333.

Daniel IO, Kruse M, Borner A (2012). "Predicting longevity of *Celosia Argentea* L. seeds during storage." *Acta Horticulturae*, **953**, 319–324.

Daniel IO, Ng NQ, Tayo TO, Togun AO (2003). "Storage of West African yam (*Dioscorea* spp.) seeds: modelling seed survival under controlled storage environments." *Seed Science and Technology*, **31**(1), 139–147.

Dehghan M, Sharif-Zadeh F (2015). "Viability model and effect of two drying procedures on seed longevity of *Secale montanum* seeds." *Biomedicine and Nursing*, **1**(1), 43–48.

Demir I, Kenanoglu BB, Hay F, Mavi K, Celikkol T (2011). "Determination of seed moisture constants (KE, CW) for the viability equation for watermelon, melon, and cucumber seeds." *Seed Science and Technology*, **39**(2), 527–532.

Demir I, Kenanoglu BB, Mavi K, Celikkol T, Hay F, Sariyildiz Z (2009). "Derivation of constants (KE, CW) for the viability equation for pepper seeds and the subsequent test of its applicability." *HortScience*, **44**(6), 1679–1682.

Dickie JB (1988). "Prospects for the long-term storage of apple seeds." *Veroffentlichungen der Landwirtschajthch-Chemischen Bundesanstalt Linz/Donau*, **19**, 47–63.

Dickie JB, Bowyer JT (1985). "Estimation of provisional seed viability constants for apple (*Malus domestica borkh.* cv. Greensleeves)." *Annals of Botany*, **56**(2), 271–275.

Dickie JB, Ellis RH, Kraak HL, Ryder K, Tompsett PB (1990). "Temperature and seed storage longevity." *Annals of Botany*, **65**(2), 197–204.

Dickie JB, May K, Morris SVA, Titley SE (1991). "The effects of desiccation on seed survival in *Acer platanoides* L. and *Acer pseudoplatanus* L." *Seed Science Research*, **1**(03).

Dickie JB, Smith RD (1995). "Observations on the survival of seeds of *Agathis* spp. stored at low moisture contents and temperatures." *Seed Science Research*, **5**(01).

Ellis RH (1988). "The viability equation, seed viability nomographs, and practical advice on seed storage." *Seed Science and Technology*, **16**(1), 29–50.

Ellis RH, Hong TD (2007). "Quantitative response of the longevity of seed of twelve crops to temperature and moisture in hermetic storage." *Seed Science and Technology*, **35**(2), 432–444.

Ellis RH, Hong TD, Roberts EH (1989). "A comparison of the low-moisture-content limit to the logarithmic relation between seed moisture and longevity in twelve species." *Annals of Botany*, **63**(6), 601–611.

Ellis RH, Hong TD, Roberts EH (1986). "Logarithmic relationship between moisture content and longevity in sesame seeds." *Annals of Botany*, **57**(4), 499–503.

Ellis RH, Hong TD, Roberts EH (1988). "A low-moisture-content limit to logarithmic relations between seed moisture content and longevity." *Annals of Botany*, **61**(4), 405–408.

Ellis RH, Hong TD, Roberts EH (1992). "The low-moisture-content limit to the negative logarithmic relation between seed longevity and moisture content in three subspecies of rice." *Annals of Botany*, **69**(1), 53–58.

Ellis RH, Hong TD, Roberts EH, Tao K (1990). "Low moisture content limits to relations between seed longevity and moisture." *Annals of Botany*, **65**(5), 493–504.

Ellis RH, Osei-Bonsu K, Roberts EH (1982). "The influence of genotype, temperature and moisture on seed longevity in chickpea, cowpea and soya bean." *Annals of Botany*, **50**(1), 69–82.

Ellis RH, Roberts EH (1980). "The influence of temperature and moisture on seed viability period in barley (*Hordeum Distichum* l.)." *Annals of Botany*, **45**(1), 31–37.

Ellis RH, Roberts EH (1981). "The quantification of ageing and survival in orthodox seeds." *Seed Science and Technology*.

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

Fantinatti JB, Usberti R (2007). "Seed viability constants for *Eucalyptus grandis*." *Pesquisa Agropecuaria Brasileira*, **42**(1), 111–117.

Ghaderi FF, Soltani A, Sadeghipour HR (2010). "Determination of seed viability constants in medicinal pumpkin (*Cucurbita pepo* L. subsp. *pepo*. convar. *pepo* var. *styriaca* Greb), borago (*Borago officinalis* L.) and black cumin (*Nigella sativa* L.)." *Journal of Plant Production (Journal of Agricultural Sciences and Natural Resources)*, **17**(3), 53–66.

Hay FR, Mead A, Manger K, Wilson FJ (2003). "One-step analysis of seed storage data and the longevity of *Arabidopsis thaliana* seeds." *Journal of Experimental Botany*, **54**(384), 993–1011.

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

Ignacio VL (2013). *Germinacao e conservacao de sementes de Balfourodendron riedelianum (Engler) Engler*. Ph.D. thesis, Universidade Estadual do Oeste do Parana.

Kebreab E, Murdoch AJ (1999). "Effect of temperature and humidity on the longevity of *Orobanche* seeds." *Weed Research*, **39**(3), 199–211.

Kraak HL, Vos J (1987). "Seed viability constants for lettuce." *Annals of Botany*, **59**(3), 343–349.

Kruse M, Ghiasi KG, Schmohl S (2005). "The seed viability equation for analysing seed storage behaviour."

Kundu M (2008). "Prediction of viability of seeds of *Pongamia pinnata* (Karanj) under controlled conditions." *Seed Science and Technology*, **36**(2), 481–485.

Kuo WHJ (1991). "On the prediction of the storage longevity of muskmelon seeds." *Memoirs of the College of Agriculture*, **31**, 22–29.

Kuo WHJ, Shan ML, Tseng MT (1990). "Effects of temperature and seed moisture-content on the longevity of sorghum seeds." *Journal of the Agricultural Association of China*, 32–41.

Lee MH, Hong SH, Na CS, Kim JG, Kim TW, Lee YH (2017). "Analysis of seed storage data and longevity for *Agastache rugosa*." *Korean Journal of Environmental Biology*, **35**, 207–214.

Muthoka PN, Hay FR, Dida MM, Nyabundi JO, Probert RJ (2009). "Moisture content and the longevity of seeds of six *Euphorbia* species in open storage." *Seed Science and Technology*, **37**(2), 383–397.

Pozitano M, Usberti R (2009). "Seed controlled deterioration of three interspecific elephant grass pearl millet hybrids." *Revista Brasileira de Zootecnia*, **38**, 428–434.

Reza EH (2014). “Estimation of seed viability constants for tall wheatgrass, cocksfoot, rye, and sheep fescue to inform gene banking decisions.” *Iranian Journal of Plant Physiology*, **4**(4), 1145–1149.

Reza EH (2014). “Prediction of seed regeneration time of some medicinal plants by estimation of viability equation constants.” *Iranian Journal of Field Crop Science*, **45**(3), 399–407.

Simoes FC, Usberti R, Paiva PDO (2008). “Controlled seed deterioration in *Tagetes patula* L. cultivars.” *Seed Science and Technology*, **36**(3), 524–533.

Sinicio R, Lopes JF, Silva DJH, Mattedi AP (2009). “Longevity equation for tomato seeds.” *Seed Science and Technology*, **37**(3), 667–675.

Tabatabaei SA (2014). “Determination of seed viability constants in sorghum under various storage conditions.” *Iranian Journal of Field Crop Science*, **45**(3), 377–387.

Tompsett PB (1984). “The effect of moisture content and temperature on the storage life of *Araucaria columnaris*.” *Seed Science and Technology*, **12**, 801–816.

Tompsett PB (1986). “The effect of temperature and moisture content on the longevity of seed of *Ulmus carpinifolia* and *Terminalia brassii*.” *Annals of Botany*, **57**(6), 875–883.

Tompsett PB (1992). “A review of the literature on storage of dipterocarp seeds.” *Seed Science and Technology*, **20**(2), 251–267.

Usberti R (2007). “Performance of tropical forage grass (*Brachiaria brizantha*) dormant seeds under controlled storage.” *Seed Science and Technology*, **35**(2), 402–413.

Usberti R, Gomes RBR (1998). “Seed viability constants for groundnut.” *Annals of Botany*, **82**(5), 691–694.

Usberti R, Roberts EH, Ellis RH (2006). “Prediction of cottonseed longevity.” *Pesquisa Agropecuaria Brasileira*, **41**(9), 1435–1441.

Wilson DO, McDonald MB (1989). “A probit planes method for analyzing seed deterioration data.” *Crop Science*, **29**(2), 471–476.

Zewdie M, Ellis RH (1991). “Response of tef and niger seed longevity to storage temperature and moisture.” *Seed Science and Technology*, **19**(2), 319–329.

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