Package ‘soilassessment’

August 30, 2020

Type Package

Title Assessment Models for Agriculture Soil Conditions and Crop Suitability

Version 0.2.1

Depends R (>= 3.5.0)

Imports arm, caret, Cubist, deSolve, e1071, FuzzyAHP, Hmisc, kernlab, maptools, methods, nnet, quantregForest, qrnn,randomForest, ranger,raster, rgdal, rpart, soiltexture, sp, stats, utils

Author Christian Thine Omuto

Maintainer Christian Thine Omuto <thineomuto@yahoo.com>

Description Soil assessment builds information for improved decision in soil management. It analyzes soil conditions with regard to agriculture crop suitability requirements [such as those given by FAO <http://www.fao.org/land-water/databases-and-software/crop-information/en/>] soil fertility classes, soil erosion models and soil salinity classification. Suitability requirements are for crops grouped into cereal crops, nuts, legumes, fruits, vegetables, industrial crops, and root crops.

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-08-30 20:00:03 UTC

R topics documented:

appendTextureclass *A function for attaching soil textural classes*

Description

This function attaches soil textural classes according to the different soil texture classification systems

Usage

```
appendTextureclass(df, method)
```

Arguments

`df`	spatial pixel dataframe with columns of soil textural proportions clay, silt, and sand in percentages
`method`	soil texture classification method for calculating soil texture. Default=USDA method

Details

df is an output of createTexturedata with spatial reference or similar dataframe with normalized proportions summing to 100 method is the texture classification method for textural class calculation. Exanple methods are USDA, FAO, Australian, German, etc.

Value

Output is a soil texture dataframe with textural classes for every row (or pixel) in the dataframe. The output may sometimes return double class such as "SaLo, Lo" implying possibility of a tie for two classes. Such outputs should be edited outside the package for meaningful representation of soil textural classes when necessary

Note

This function can sometimes return double classes such as "SaLo, Lo" implying possibility of a tie for two classes.

Author(s)

Christian Thine Omuto

References

Moyes J. 2018. The soil texture wizard: R functions for plotting, classifying, transforming and exploring soil texture data. https://cran.r-project.org/web/packages/soiltexture/vignettes/soiltexture_vignette.pdf

See Also

textureclassLUT, textureLevels, createTexturedata

Examples

```
library(soiltexture)
newtxt=textureinput
texturedata=createTexturedata(newtxt$clay, newtxt$silt, newtxt$sand)
newtxt1=appendTextureclass(as.data.frame(texturedata), method = "USDA")
levels(as.factor(newtxt1$TEXCLASS))
```

---

carbonateSuit *A function for assessing calcium carbonate suitability requirements for certain crops*

Description

This function determines the suitability classes for soil calcium carbonate requirements of selected agricultural crops

Usage

```
carbonateSuit(value, crop)
```

Arguments

`value`	Input calcium carbonate content in percent
`crop`	The crop of interest for which calcium carbonate suitability class is sought

Details

The input value can be map or just a numerical entry of calcium carbonate in percent

Value

The output is calcium carbonate suitability class for the crop. The output is an integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

depthSuit, SOCSuit, suitability

Examples

```
library(sp)
newmap=suitabinput
newmap$saffron=carbonateSuit(newmap$cac03,"saffron")
summary(newmap$saffron)
spplot(newmap,"saffron")
```

---

carbonTurnover *A function for implementing RothC carbon turnover model in the soil*

Description

This function provides alternatives for implementing RothC carbon turnover model

Usage

```
carbonTurnover(tt,clay,C0,In,Dr=1.44,effcts,solver)
```

Arguments

`tt`	a vector of time in months or years for modelling carbon turnover in the soil
`clay`	Proportion of soil clay content in percent
`C0`	a vector containing five initial carbon pools in the five compartments C1 in DPM, C2 in RPM, C3 in BIO, C4 in HUM and C5 in IOM. They are arranged in the order C1,C2,C3,C4,C5 [C0=c(C1,C2,C3,C4,C5)].
`In`	Input carbon amount. It can be a scalar constant or a 2-column dataframe containing time dependent organic matter input. The two columns are time and carbon input
`Dr`	ratio of decomposable plant material (DPM) to resistant plant material (RPM). Default value is 1.44
`effcts`	a constant or dataframe of environmental effects on carbon decomposition rates. If it's a dataframe of time-dependent variables, then the length of the dataframe should be similar to the length of time (t) vector

solver	name of subroutines for solving first order odinary differential equations for organic matter decay in the soil. The subroutines are lsoda, lsodes,rk4, euler,lsode, lsodar,ode23, radau,etc. from deSolve

Details

vector t can be years or months sequentially arranged with the start-time as the minimum and end-time as the maximum time. Initial carbon pools are also provided as a vector of five items: C1,C2,C3,C4,C5 in that order where C1 is the pool in the decomposable plant material (DPM) compartment, C2 is pool in the resistant plant material (RPM) compartment, C3 is the pool in the microbial biomass (BIO) compartment, C4 is the pool in the humified organic matter (HUM) compartment, and C5 is the pool in the inert organic matter (IOM) compartment.

Value

nx6 matrix of carbon pools with time in the five compartments DPM, RPM, BIO, HUM, and IOM in that order (time, C1, C2, c3, C4, C5).

Author(s)

Christian Thine Omuto

References

Coleman, K. and Jenkinson, D. 2014. ROTHC-26.3 A model for the turnover of carbon in soils: Model description and users guide (Windows version). Rothamsted Research Harpenden Herts AL5 2JQ

Jenkinson, D. S., Andrew, S. P. S., Lynch, J. M., Goss, M. J., Tinker, P. B. 1990. The Turnover of Organic Carbon and Nitrogen in Soil. Philosophical Transactions: Biological Sciences, 329: 361–368.

See Also

RotCmoiscorrection, RotCtempcorrection, NPPmodel

Examples

```
library(deSolve)
Cin=c(0.6,0.1,0.3,0.1,2.7)
T=seq(1/12,200,by=1/12)
hw=carbonTurnover(tt=T,clay=23.4,C0=Cin,In=1.2,Dr=1.44,effcts=0.85,"euler")
matplot(T,hw[,2:6], type="l", lty=1, xlab="Time", ylab="C stocks (Mg/ha)")
legend("topright", c("DPM", "RPM", "BIO", "HUM", "IOM"),lty=1, col=1:5, bty="n")
```

`CECSuit` *A function for assessing Cation Exchange Capacity (CEC) suitability requirements for certain crops*

Description

This function determines the suitability classes for Cation Exchange Capacity (CEC) requirements of selected agricultural crops

Usage

```
CECSuit(value, crop)
```

Arguments

`value`	Input cation Exchange Capacity (Cmol(+)/kg).
`crop`	The crop of interest for which CEC suitability class is sought

Details

The input value can be a map or just a numerical entry of CEC (cmol(+)/kg)

Value

The output is CEC suitability class for the crop. The output is an integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

The output raster map of CEC suitability is given if the input value is raster map

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, PHSuit, rainSuit

Examples

```
CECSuit(22.4,"pineaple")
```

`classCode` *A function for displaying names of class codes of soil conditions in the soilassessment package*

Description

This function displays names of integer classes (or levels) of derived codes of soil conditions produced in the package

Usage

```
classCode(value, indicator)
```

Arguments

`value`	Integer value of the soil condition indicator
`indicator`	Soil condition whose class (x) is sought. The default = "fertility" if fertility is the soil condition

Details

This is for interpretation of the integer codes of the soil conditions generated in the package

Value

Name of the level of soil condition

See Also

classLUT, erodFUN, classnames

Examples

```
classCode(2,"texture")
suitclas=classCode(4,"suitability")
levels(suitclas)
```

`classLUT` *A function for developing Look-up Table (LUT) for the soil condition class map*

Description

This function develops a Look-Up Table (LUT) for the class type map of soil condition. LUT is important map legends or maps re-classification.

Usage

```
classLUT(fgrid,indicator)
```

Arguments

`fgrid`	Input classified map
`indicator`	The soil condition indicator of interest as contained in the input map for example, "texture", "salinity", etc.

Details

The input raster map should contain only one band for the soil indicator for clear identification of the band.

Value

The output is a dataframe containing classes in the map and corresponding unique integers

Author(s)

Christian Thine Omuto

See Also

classCode, classnames

Examples

```
textrd=suitabinput["texture"]

LUT=classLUT(textrd,"texture")
LUT
```

classnames — *A function to display the class names and codes as used in the soilassessment package*

Description

This is a database function for displaying the class names and codes used in the soil assessment package

Usage

```
classnames(indicator)
```

Arguments

`indicator`	indicator of soil condition group of interest. Example: texture, suitability, drainage, fertility, erodibility

Value

Table of soil condition code and name

See Also

classCode, classLUT

Examples

```
x="texture"
classnames(x)
```

comparisonTable — *A function for normalizing decision ranking table*

Description

This function normalizes the decision ranking table and determines consistency of the decisions

Usage

```
comparisonTable(df)
```

Arguments

`df`	A matrix of rank decisions with complete column names.

Details

The column names of the rank-decision table should correspond with the names of the criteria maps

Value

nmtx: a normalized pairwise comparison matrix crt: consistency index and message on whether the input decisions are consistent for analysis

Author(s)

Christian Thine Omuto

References

Barzilai J. and Golany B., 1990. Deriving Weights from Pairwise Comparison Matrices: the Additive Case. Operations Research Letters 9: 407–410.

See Also

suitability

Examples

```
data(nutrient)
library(FuzzyAHP)
comparisonTable(nutrient)
```

`createTexturedata` *A function for creating spatial dataframe of normalized soil texture proportions*

Description

The function creates spatial dataframe of normalized soil texture proportions. They are normalized to 100 percent

Usage

```
createTexturedata(clay,silt,sand)
```

Arguments

`clay`	clay proportion of soil texture in percent
`silt`	silt proportion of soil texture in percent
`sand`	sand proportion of soil texture in percent

Details

the input data of soil texture proportions are imported into R as spatial raster or dataframe. They need to have uniform coordinate reference system (CRS) and same pixel size (resolution) if in raster map format. The sum of the proportions should be close to 100 per cent for each row

Value

The output is a spatial pixel dataframe of normalized soil texture proportions (for each pixel)

Note

It's important to ensure the input data does not have negative values nor add up to far below or above 100 per cent. It's also important to adhere to the order of the input data: clay, silt, sand

Author(s)

Christian Thine Omuto

See Also

classLUT, appendTextureclass, createTexturedata

Examples

```
#data(textureinput)
newmap=textureinput

texturedata=createTexturedata(newmap$clay, newmap$silt, newmap$sand)
cor(texturedata$CLAY,texturedata$CLAY_n)^2
```

`depthSuit` *A function for assessing soil depth suitability requirements for certain crops*

Description

This function determines the suitability classes for soil depth requirements of selected agricultural crops

Usage

```
depthSuit(value, crop)
```

Arguments

`value`	Input soil depth in cm
`crop`	The crop of interest for which soil depth suitability class is sought

Details

The input value can be map or just a numerical entry of soil depth in cm

Value

The output is soil depth suitability class for the crop. The output is integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, ESPSuit, classCode

Examples

```
#data(suitabinput)
library(sp)
library(raster)
LUT=data.frame(map=c(1,2,3,4,5,6),new=c(100,20,30,40,60,80))
newmap=(suitabinput["depthcodes"])
newmap$depth=reclassifyMap(newmap["depthcodes"],LUT)
newmap$melon=depthSuit(newmap$depth,"melon")
summary(newmap$depth)
spplot(newmap["depth"])
```

---

drainageSuit — *A function for assessing drainage suitability requirements for certain crops*

Description

This function determines the suitability classes for drainage requirements for selected agricultural crops

Usage

```
drainageSuit(value, crop)
```

Arguments

`value`	Input drainage class code
`crop`	The crop of interest for which drainage suitability class is sought.

Details

The input value can be a map or an integer of drainage class code. The textural class code is obtained using classCode("drainage")

Value

The output is drainage suitability class for the crop. The output is an integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

If the input value is raster map, then the output will also be a raster map of drainage suitability for the crop of interest

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, PHSuit, rainSuit

Examples

```
drainageSuit(6,"cassava")
```

`ECconversion1` *A function for correcting electrical conductivity of soil solution to that of saturated paste extract*

Description

This function converts electrical conductivity measurements of a soil solution to that of soil paste extract. It considers the influence of texture, organic matter content, and clay content on electrical conductivity conversion. These factors and ratio of soil:water mix for the solution and conversion method must be indicated.

Usage

```
ECconversion1(ec,oc,clay,texture,soilsolution, method)
```

Arguments

`ec`	measured electrical conductivity of the soil solution in dS/m
`oc`	organic matter content of the soil in percent
`clay`	clay content of the soil in percent
`texture`	soil textural class according to USDA or its equivalent. Texture class is given in terms of class codes as given in classnames("texture")
`soilsolution`	ratio of soil:water mix when electrical conductivity was measured. Example is 1:1, 1:2, etc. The default is 1:1
`method`	method for converting electrical conductivity of the soil:water mix to that of the soil paste extract. The methods included are FAO, sonmez, and hogg. The default is FAO

Details

This function considers the influence of texture, clay content, organic matter content, and soil-water solution on conversion of electrical conductivities

Value

electrical conductivity equivalent for saturated soil extract in dS/m

Author(s)

Christian Thine Omuto

References

FAO. 2006. Soil description guidelines. FAO, Rome.

Sonmez S, Buyuktas D, Asri FO. 2008. Assessment of different soil to water ratios (1:1, 1:2.5, 1:5) in soil salinity studies. Geoderma, 144: 361-369

Hogg TTJ, Henry JL. 1984. Comparison of 1:1 and 1:2 suspensions and extracts with the saturation extracts in estimating salinity in Saskatchewan. Can. J. Soil Sci. 1984, 64, 699–704

See Also

ECconversion2, saltRating, saltClass, saltSeverity

Examples

```
library(sp)
library(raster)
ECconversion1(7.31,1.02,22.1,5,"1:2.5", "FAO")
ec=suitabinput["ec"]
soc=nutrindicator["soc"]
clay=textureinput["clay"]
texture=suitabinput["texture"]
newmap=ec
newmap$ECe=ECconversion1(ec$ec,soc$soc,clay$clay,texture$texture,"1:2.5", "FAO")
spplot(newmap["ECe"], main="Equivalent ECse")
```

`ECconversion2` *A function for correcting electrical conductivity of soil solution to that of saturated paste extract for all textural classes*

Description

This function converts electrical conductivity measurements of soil solution to that of soil paste extract. The ratio of soil:water mix for the solution and conversion method must be indicated

Usage

```
ECconversion2(ec, soilsolution, method)
```

Arguments

`ec`	measured electrical conductivity of the soil solution in dS/m
`soilsolution`	ratio of soil:water mix when electrical conductivity was measured. Example is 1:1, 1:2, etc. The default is 1:1
`method`	method for converting electrical conductivity of the soil:water mix to that of the soil paste extract. The methods included are USDA, landon, kargas, ozkan, and hogg. The default is USDA

Details

This function assumes no influence of texture, clay content, etc on the conversion of electrical conductivities

Value

electrical conductivity equivalent for saturated soil extract in dS/m

Author(s)

Christian Thine Omuto

References

Sonmez S, Buyuktas D, Asri FO. 2008. Assessment of different soil to water ratios (1:1, 1:2.5, 1:5) in soil salinity studies. Geoderma, 144: 361-369

Kargas G, Chatzigiakoumis I, Kollias A, Spiliotis D, Massas I, Kerkides P. 2018. Soil salinity assessment using saturated paste and mass soil:water 1:1 and 1:5 ratios extracts. Water, 10:1589, doi:10.3390/w10111589

See Also

ECconversion1, saltClass, saltRating

Examples

```
library(sp)
ECconversion2(0.75,"1:1", "USDA")
newmap = suitabinput["ec"]
newmap$salinity=ECconversion2(newmap$ec,"1:1","hogg")
str(newmap$salinity)
spplot(newmap["salinity"])
```

ECconversion3 *A function for developing own conversion model*

Description

This function enables development of own function for converting soil assessment indicators to those determined from the standard saturated soil paste

Usage

```
ECconversion3(x, A, B, method)
```

Arguments

x	input predictor value
A	A location parameter representing the value of target variable when the predictors are minimal (or the y-intercept)
B	Rate parameter representing the rate of change of the target variable with the predictor (or the slope)
method	model relationship between target and predictor variables

Details

model for the relationship between target and predictor variables can be "linear", "power", "exponential", "log". Default is "linear"

Value

model object containing predictive parameters of the conversion model

Author(s)

Christian Thine Omuto

References

van Looy k, Bouma J, Herbst M, Koestel J, Minasny B, Mishra U, Montzka C, Nemes A, Pachepsky AY, Padarian J, Schaap MG, Tóth B, Verhoef A, Jan Vanderborght, van der Ploeg MJ, Weihermüller L, Zacharias S, Zhang Y, Vereecken H. 2017. Pedotransfer functions in Earth System Science: Challenges and Perspectives. Reviews of Geophysics 55(4): 1199-1256.

Sudduth KA, Kitchen RN, Wiebold WJ, Batchelor W. 2005. Relating apparent electrical conductivity to soil properties across the North-Central USA. Computers and Electronics in Agriculture, 46(1-3):263-283

See Also

ECconversion1, ECconversion2, saltRating

Examples

```
x=as.vector(c(0.800,2.580,0.980,0.532,1.870, 18.500,0.430,0.302,0.345,2.700))
y=as.vector(c(17.88,  6.43,  3.83,  7.18,  6.64, 14.83,  4.19,  7.31,  3.21, 18.41))
xy=as.data.frame(cbind(x,y))
names(xy)=c("ECa", "EC")
EC3.ml=nls(EC~ECconversion3(ECa,A,B), start=c(A=0.1, B=0.8), data=xy)
cor.test(fitted(EC3.ml),xy$EC)
plot(fitted(EC3.ml)~xy$EC)
abline(0,1)
```

ECconversion4 — *A function for converting salt measurements into equivalent electrical conductivity in dS/m*

Description

This function allows approximate conversion of other soil salt measurements into equivalent electrical conductivity (EC) in dS/m. These measurements include total soluble salts (TSS), total dissolved solids (TDS) and EC in mmho/cm

Usage

```
ECconversion4(x,target)
```

Arguments

`x`	is a numeric value of salt to convert to equivalent EC in dS/m
`target`	the target salt measurement to be converted into equivalent electrical conductivity (EC) in dS/m. It can be TDS (mg/l or ppm), TSS (mmol/l), EC in (mmho/cm)

Details

The target is specified as TDS or TSS or mmho or Sodium Adsorption Ratio (SAR).

Value

The output is a numeric value of equivalent electrical conductivity (EC) in dS/m

Note

TDS should be given in mg/l or ppm. TSS should be given in mmol/l. The function does not convert salt values between different measurement methods

Author(s)

Christian Thine Omuto

See Also

ECconversion1, ECconversion2, ECconversion3, pedoTransfer

Examples

```
ECconversion4(200,"TSS")
ECconversion4(20,"TDS")
ECconversion4(2,"mmho")
ECconversion4(10.4,"SAR")
```

`ECSuit` *A function for assessing Electrical Conductivity suitability requirements for certain crops*

Description

This function determines the suitability classes for Electrical Conductivity requirements for selected agricultural crops

Usage

```
ECSuit(value, crop)
```

Arguments

`value`	Input electrical conductivity in dS/m.
`crop`	The crop of interest for which Electrical Conductivity suitability class is sought.

Details

The input value can be map or just a numerical entry of electrical conductivity (ECe) of saturated paste extract or its equivalent in dS/m

Value

The output is EC suitability class for the crop. The output is integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

If the input value is raster map, then the output will also be a raster map of Electrical Conductivity suitability for the crop of interest

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, PHSuit, rainSuit

Examples

```
library(sp)
ECSuit(0.78,"yam")
ec=(suitabinput["ec"])
soc=(nutrindicator["soc"])
clay=(textureinput["clay"])
texture=(suitabinput["texture"])
newmap=ec
newmap$ECe=ECconversion1(ec$ec,soc$soc,clay$clay,texture$texture,"1:2.5", "FAO")
newmap$wheat=ECSuit(newmap$ECe,"wheat")
spplot(newmap["wheat"], main="EC suitability for wheat")
summary(newmap$wheat)
```

`erodFUN` *A function to estimate soil erodibility factor*

Description

A function to determine soil erodibility factor from a choice of different erodibility models

Usage

```
erodFUN(sand,silt,clay,OC,texture,Struct,method)
```

Arguments

`sand`	sand proportion (percent) of the soil texture
`silt`	silt proportion (percent) of the soil texture
`clay`	clay proportion (percent) of the soil texture
`OC`	soil carbon content (percent)
`texture`	soil texture code representing the USDA soil textural class. Use classnames("texture") to insert the correct texture code
`Struct`	soil structure code representing the soil structure class. Use classnames("structure") to insert the correct structure code
`method`	method for determining soil erodibility. The following methods are included: WSmith,Yang,Renard,Bouyoucos,Denardin,Wang,Wisch1,Wisch2,Sharpley,Cheng,Auer.

Value

soil erodibility factor ranging between 0 and 1

Author(s)

Christian Thine Omuto

References

Benavidez R, Bethana J, Maxwell D, Norton K. 2018. A review of the (Revised) Universal Soil Loss Equation ((R)USLE): with a view to increasing its global applicability and improving soil loss estimates. Hydrol. Earth Syst. Sci., 22, 6059–6086

Omuto CT and Vargas R. 2009. Combining pedometrics, remote sensing and field observations for assessing soil loss in challenging drylands: a case study of northwestern Somalia. Land Degrad. Develop. 20: 101–115

See Also

erosivFUN, erodibilityRisk, slopelenFUN, permeabilityClass, classCode, classnames

Examples

```
library(sp)
bx=suitabinput
sand=textureinput["sand"]
silt=textureinput["silt"]
clay=textureinput["clay"]
soc=nutrindicator["soc"]
bx$permeability=permeabilityClass(bx$texture)
bx$wsmith=erodFUN(sand$sand,silt$silt,clay$clay,soc$soc,bx$texture, bx$structure,"WSmith")
bx$renard=erodFUN(sand$sand,silt$silt,clay$clay,soc$soc,bx$texture, bx$structure,"Renard")
summary(bx$renard)
spplot(bx["wsmith"])
```

---

erodibilityRisk — *A function to determine soil erodibility risk*

Description

This function classifies soil erodibility factor into classes of risk to erosion

Usage

```
erodibilityRisk(x)
```

Arguments

x	soil erodibility factor value between 0 and 1

Details

Erodibility factor ranges between 0 (lowest risk) to 1 (highest risk)

Value

erodibility risk classes

Author(s)

Christian Thine Omuto

References

Wischmeier WH, Mannering JV. 1969. Relation of Soil Properties to its Erodibility, Soil and Water Management and Conservation, 15, 131–137 Benavidez R, Bethana J, Maxwell D, Norton K. 2018. A review of the (Revised) Universal Soil Loss Equation ((R)USLE): with a view to increasing its global applicability and improving soil loss estimates. Hydrol. Earth Syst. Sci., 22, 6059–6086

See Also

erosivFUN, erodFUN, slopelenFUN

Examples

```
library(sp)
erodibilityRisk(0.8)
x=suitabinput
sand=textureinput["sand"]
silt=textureinput["silt"]
clay=textureinput["clay"]
soc=nutrindicator["soc"]
x$permeability=permeabilityClass(x$texture)
x$renard=erodFUN(sand$sand,silt$silt,clay$clay,soc$soc,x$texture, x$structure,"Renard")
x$erodibilityrisk=erodibilityRisk(x$renard)
x$erodib=classCode(x$renard,"erodibility")
summary(x$erodib)
spplot(x["erodib"])
```

---

erosivFUN *A function to estimate rainfall erosivity from annual rainfall amounts*

Description

This function assumes an algebraic relationship between annual rainfall amounts and rainfall erosivity. The relationship has constants that may depend of certain regions.

Usage

```
erosivFUN(rain,A,B, model)
```

Arguments

`rain`	annual rainfall amounts in mm or Fourier index of rainfall
`A`	independent constant of the algebraic relationship between rainfall mounts and erosive energy (Energy=A+-B*rainfall)
`B`	rainfall coefficient of the algebraic relationship between rainfall mounts and erosive energy (Energy=A+-B*rainfall)
`model`	model defining the algebraic relationship between rainfall mounts and erosive energy. The model can be linear, power, logarithmic, Fourier, and exponential

Value

rainfall erosivity in MJ mm/ha/hr/yr

Author(s)

Christian Thine

References

Morgan RPC. 2005. Soil erosion and conservation. Blackwell. UK Benavidez R, Bethana J, Maxwell D, Norton K. 2018. A review of the (Revised) Universal Soil Loss Equation ((R)USLE): with a view to increasing its global applicability and improving soil loss estimates. Hydrol. Earth Syst. Sci., 22, 6059–6086

See Also

erodFun, sloplenFun

Examples

```
erosivFUN(587,151, 0.63, "linear")
```

`ESPSuit` *A function for assessing Exchangeable Sodium Percent (ESP) suitability requirements for certain crops*

Description

This function determines the suitability classes for ESP requirements of selected agricultural crops

Usage

```
ESPSuit(value, crop)
```

Arguments

`value`	Input Exchangeable Sodium Percent (ESP).
`crop`	The crop of interest for which ESP suitability class is sought.

Details

The input value can be map or just a numerical value of Exchangeable Sodium Percent (ESP).

Value

The output is ESP suitability class for the crop. The output is integer value for suitability class: 1-highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

If the input value is raster map, then the output will also be a raster map of ESP suitability for the crop of interest

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, PHSuit, rainSuit

Examples

```
ESPSuit(8.6,"broccoli")
```

---

featureRep *A function to assess how well landscape features have been sampled*

Description

This function establishes graphical representation of the landscape feature in the sample points. An approximation of Kolmogorov-Smirnov similarity test (D-statistic) between the sampled feature distribution and the population feature distribution is also given.

Usage

```
featureRep(fgrid,df )
```

Arguments

`fgrid`	raster grid of the landscape feature
`df`	dataframe of sampled locations with similar coordinate reference system (CRS) as the input raster map

Details

The sampled points should have the same coordinate system as the landscape feature (raster map). The function extracts the raster map values, attaches them to the sample points, and creates histogram distributions: one for the feature map as contained in the sample points and another as contained in the raster map.

Value

Histograms on back-to-back showing distribution of the landscape feature in the sampled points and on the map for similarity comparison

Note

The input points dataframe and raster map must have similar coordinate reference system.

Author(s)

Christian Thine Omuto

References

Kolmogorov, A. N. 1933. Sulla determinazione empirica di una legge di distribuzione. Giornale dell' Istituto Italiano degli Attuari 4: 83–91

Simard R, L'Ecuyer P. 2011. Computing the Two-Sided Kolmogorov–Smirnov Distribution. Journal of Statistical Software. 39 (11): 1–18. doi:10.18637/jss.v039.i11

See Also

imageIndices, regmodelSuit

Examples

```
library(Hmisc)
data(soil)
dem=suitabinput["dem"]
featureRep(dem,soil)
```

`fertilityRating` *A function for determining soil fertility levels for given soil property (fertility indicator)*

Description

This function determines the fertility levels given values of a soil property

Usage

```
fertilityRating(value, indicator = "nitrogen")
```

Arguments

`value`	numerical value of soil property
`indicator`	soil property as fertility indicator

Details

The units for input values are: nitrogen (percent), phosphorus (mg/kg); potassium (cmol(+)/kg);carbon(percent);iron(mg/kg);
CEC(cmol(+)/kg)

Value

soil fertility class code for the given soil property (fertility indicator)

Author(s)

Christian Thine Omuto

References

FAO, 1976. A framework for land evaluation. FAO Soils Bulletin 32 Sanchez PA, Couto W, Buol SW. 1982. The fertility capability soil classification system: Interpretation, applicability, and modification

Sanchez PA, Palm CA, Buol SW. 2003. Fertility capability soil classification: a tool to help assess soil quality in the tropics. Geoderma 114, 157 –185.

See Also

ECconversion1,ECconversion2, saltRating,suitability

Examples

```
library(sp)
newmap=nutrindicator["iron"]
newmap$ironclass=fertilityRating(newmap$iron,"iron")
summary(newmap$iron)
spplot(newmap["ironclass"])
```

`fertilitySuit` *A function for assessing soil fertility suitability requirements for certain crops*

Description

This function determines the suitability classes for soil fertility requirements of selected agricultural crops

Usage

```
fertilitySuit(value, crop)
```

Arguments

`value`	Input soil fertility index.
`crop`	The crop of interest for which soil fertility suitability class is sought.

Details

The input value can be map or just a numerical entry of soil fertility index

Value

The output is fertility suitability class for the crop. The output is integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

If the input value is raster map, then the output will also be a raster map of fertility suitability for the crop of interest

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, PHSuit, rainSuit

Examples

```
library(sp)
library(FuzzyAHP)
fertilitySuit(1.56, "melon")
newmap=(nutrindicator)
newmap$carbon=fertilityRating((nutrindicator$soc),"carbon")
newmap$nitrogen=fertilityRating((nutrindicator$nitrogen),"nitrogen")
newmap$potassium=fertilityRating((nutrindicator$potassium),"potassium")
newmap$phosphorus=fertilityRating((nutrindicator$phosphorus),"phosphorus")
newmap$iron=fertilityRating((nutrindicator$iron),"iron")
newmap$zinc=fertilityRating((nutrindicator$zinc),"zinc")
newmap$manganese=fertilityRating((nutrindicator$manganese),"manganese")
newmap$copper=fertilityRating((nutrindicator$copper),"copper")
newmap$cec=fertilityRating((nutrindicator$cec),"cec")
newmap$boron=fertilityRating((nutrindicator$boron),"boron")
newmap$sulfur=fertilityRating((nutrindicator$sulfur),"sulfur")
newmap$soc=NULL
newmapT1=newmap@data
valuT=as.matrix(newmapT1)
data("nutrient")
nutriens=comparisonTable(nutrient)

newmapT1$fertility=suitability(nutrient, valuT)
newmap@data$fertility=newmapT1$fertility
newmap$fertilityokra=fertilitySuit(newmap$fertility,"okra")
str(newmap$fertilityokra)
spplot(newmap["fertilityokra"], main="Fertility suitability map for Okra")
```

`imageIndices` *A function for developing remote sensing indices for soil assessment*

Description

The function determines commonly used remote sensing indices with relationship with soil surface or vegetation cover characteristics.

Usage

```
imageIndices(blue, green,red,nir,swir1,swir2,index)
```

Arguments

`blue`	blue image band with wavelength range: 0.452-0.512 µm
`green`	green image band with wavelength range: 0.533-0.59 µm
`red`	red image band with wavelength range: 0.636-0.673 µm
`nir`	NIR image band with wavelength range: 0.851-0.879 µm
`swir1`	SWIR image band with wavelength range: 1.566-1.651 µm
`swir2`	SWIR image band with wavelength range: 2.107-2.294 µm
`index`	index from combination of image bands such as NDVI, SAVI, SI, etc. The default is NDVI.

Details

The indices included in the package are: NDVI, NDSI (for salinity), NDSnI (for snow), SI1, SI2, SI3, SI4, SI5, SI6, SAVI, VSSI, NDSI, SR, CRSI, BI, ROCK

Value

dimensionless remote sensing index

Author(s)

Christian Thine Omuto

References

Gorji T, Yildirim A, Sertel E, Tanik A. 2019. Remote sensing approaches and mapping methods for monitoring soil salinity under different climate regimes. International Journal of Environment and Geoinformatics 6(1): 33-49 (2019)

See Also

featureRep

Examples

```
imageIndices(0.15,0.05,0.18,0.25,0.36,0.45,"SAVI")
```

`LGPSuit` *A function for assessing Length of Growing Period (LGP) suitability requirements for certain crops*

Description

This function determines the suitability classes for length of growing period (LGP) requirements for selected agricultural crops

Usage

```
LGPSuit(value, crop)
```

Arguments

`value`	Input length of growing period (LGP) in days.
`crop`	The crop of interest for which length of growing period (LGP) suitability class is sought.

Details

The input value can be map or an integer value of LGP in days

Value

The output is LGP suitability class for the crop. The output is an integer for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

If the input value is raster map, then the output will also be a raster map of LGP suitability for the crop of interest

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, PHSuit, rainSuit

Examples

```
library(sp)
library(maptools)
LGPSuit(138,"cotton")
newmap = data.frame(LGP = c(1:6,158,160,211),
                    lon = c(1,1,1,2,2,2,3,3,3),
                    lat = c(rep(c(0, 1.5, 3),3)))
coordinates(newmap) = ~lon+lat
gridded(newmap) = TRUE
newmap = as(newmap, "SpatialGridDataFrame")
newmap$LGPmillet=LGPSuit(newmap$LGP,"millet")
spplot(newmap["LGPmillet"], main="LGP suitability map for finger millet")
```

NPPmodel — *A function for calculating net primary production using air temperature and mean rainfall amount*

Description

This is an empirical function for deriving net primary production using climatic variables (mean temperature and rainfall amounts)

Usage

```
NPPmodel(rain,temperature,model)
```

Arguments

`rain`	total annual rainfall amount in mm
`temperature`	average annual air temperature amount in degrees Celsius
`model`	model for calculating net primary production. Included models in the function are Miami, Schurr, and NCEAS

Details

This function is based on empirical models for calculating annual net primary production (NPP) of dry matter

Value

Net primary production (NPP) of dry matter in grams per square meter per year

Note

This empirical function estimates annual NPP in g/m2/year. It is a general model for all land cover types. It may be necessary to adjust it for certain cover types or geolocations

Author(s)

Christian Thine Omuto

References

Schuur, E. A. G. 2003. Productivity and global climate revisited: the sensitivity of tropical forest growth to precipitation. Ecology 84:1165–1170

Lieth, H. 1975. Modeling the primary productivity of the world. Pages 237–264 in H. Lieth and R. H. Whittaker, editors. Primary productivity of the biosphere. Springer-Verlag, New York, New York, USA

Del Grosso, S., Parton, W., Stohlgren, T., Zheng, D., Bachelef, D., Prince, S., Hibbard, K., Olson, R. 2008. Global potential net primary production predicted from vegetation class, precipitation, and temperature. Ecology, 89(8): 2117-2126

See Also

RotCmoistcorrection, RotCtempcorrection, carbonTurnover

Examples

```
NPPmodel(800,23,"miami")
NPPmodel(800,23,"schuur")
NPPmodel(800,23,"NCEAS")
```

`nutrient` *Sample data of decision ranking table for mapping soil nutrient condition*

Description

This is an 11-factor table of decision ranking of soil nutrient indicators

Usage

```
data("nutrient")
```

Format

A dataframe with 11 factors for pairwise decision ranking of soil nutrient indicators.

Details

The ranks are the recipricals of Saaty's scale of relative importance which are between 1 and 9

Source

Hypothetical data of rank between soil nutrient indicators

Examples

```
data(nutrient)
str(nutrient)
plot(nutrient)
```

`nutrindicator` *A grid stack map of indicators for crop fertility requirements*

Description

A grid stack map of eleven variables for assessing soil fertility

Usage

```
data("nutrindicator")
```

Format

Formal class 'SpatialGridDataFrame' [package "sp"] with 4 slots ..@ data :'data.frame': 16900 obs. of 11 variables: .. ..$ soc : num [1:16900] 0.163 0.242 0.233 0.218 0.179 ... .$ nitrogen : num [1:16900] 0.0272 0.0242 0.0266 0.0275 0.0256 ... .$ phosphorus: num [1:16900] 9.4 8.22 8.92 7.45 8.3 ... .$ manganese : num [1:16900] 2.84 2.7 2.95 2.88 3.19 ... .$ potassium : num [1:16900] 93.2 102.3 93.5 96.5 87.8 ... .$ cec : num [1:16900] 10.9 10.7 10 10.1 10.2 ... .$ boron : num [1:16900] 0.172 0.16 0.171 0.172 0.174 ... .$ copper : num [1:16900] 0.368 0.421 0.37 0.369 0.412 ... .$ iron : num [1:16900] 0.238 0.231 0.241 0.239 0.242 ... .$ zinc : num [1:16900] 0.816 0.652 0.816 0.818 0.814 ... .$ sulfur : num [1:16900] 153 131 119 135 163 ... ..@ grid :Formal class 'GridTopology' [package "sp"] with 3 slots .. .. ..@ cellcentre.offset: Named num [1:2] 383216 3341506 .. ..- attr(*, "names")= chr [1:2] "x" "y" .. .. ..@ cellsize : num [1:2] 357 357 .. .. ..@ cells.dim : int [1:2] 130 130 ..@ bbox : num [1:2, 1:2] 383038 3341327 429478 3387767 .. ..- attr(*, "dimnames")=List of 2 .. .. ..$: chr [1:2] "x" "y" .. .. ..$: chr [1:2] "min" "max" ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot .. .. ..@ projargs: chr "+proj=utm +zone=41 +datum=WGS84 +units=m +no_defs"

Examples

```
data(nutrindicator)
str(nutrindicator)
#spplot(nutrindicator["nitrogen"])
```

pedoTransfer — *A pedotransfer function to predict electrical conductivity or any other soil property using other soil properties*

Description

This generic pedo-transfer function is used to approximate EC values from other existing and easy-to-measure soil data

Usage

```
pedoTransfer(method="linear", df, ...)
```

Arguments

`method`	modelling method to link EC and other soil predictors (properties). Default method is linear
`df`	dataframe containing measured EC and predictors of soil properies
`...`	names of measured EC and list of predictors (soil properties) seperated by comma. The names should match the variables in the accompanying dataframe

Details

This generic model can be used even with other soil properties. For example, it can be used to predict porosity from bulk density, carbon, and texture components as long as they are in the database and have known/suspected relationship

Value

model for predicting EC given similar input data

Note

This function can also be used to predict EC from apparent electrical conductivity of bulk soil, texture, and other important soil properties

Author(s)

Christian Thine Omuto

References

van Looy k, Bouma J, Herbst M, Koestel J, Minasny B, Mishra U, Montzka C, Nemes A, Pachepsky AY, Padarian J, Schaap MG, Tóth B, Verhoef A, Jan Vanderborght, van der Ploeg MJ, Weihermüller L, Zacharias S, Zhang Y, Vereecken H. 2017. Pedotransfer functions in Earth System Science: Challenges and Perspectives. Reviews of Geophysics 55(4): 1199-1256.

Sudduth KA, Kitchen RN, Wiebold WJ, Batchelor W. 2005. Relating apparent electrical conductivity to soil properties across the North-Central USA. Computers and Electronics in Agriculture, 46(1-3):263-283

See Also

ECconversion1, ECconversion2, ECconversion4

Examples

```
library(caret)
clay=as.data.frame(runif(120, 1,100))
silt=as.data.frame (runif(120,20,70))
sand=as.data.frame(runif(120,10.1,50.5))
pH=as.data.frame(runif(120,1,14))
EC=as.data.frame(runif(120,0.5,20.5))
OC=as.data.frame(runif(120,0.1,1.25))
soil4=cbind(EC,clay,silt,sand,OC,pH)
names(soil4)=c("EC","clay","silt","sand","OC","pH")
bound <- floor((nrow(soil4)/4)*3)
df.train <- soil4[sample(nrow(soil4)), ][1:bound, ]
df.test <- soil4[sample(nrow(soil4)), ][(bound+1):nrow(soil4[sample(nrow(soil4)), ]), ]
EC1.lm=pedoTransfer("randomforest",df.train,EC, clay,sand,silt,OC,pH)
df.test$EC1=predict(EC1.lm,newdata = df.test)
cor(df.test$EC,df.test$EC1)^2
plot(df.test$EC~df.test$EC1)
abline(1,1)
```

---

permeabilityClass — *A function to determine permeability class*

Description

This function determines the soil permeability classes according to the USDA soil textural classes

Usage

```
permeabilityClass(texture)
```

Arguments

`texture`	soil textural class code

Details

Soil textural class code is obtained using classnames("texture")

Value

permeability class code

Author(s)

Christian Thine Omuto

References

O'Geen, A. T. (2013) Soil Water Dynamics. Nature Education Knowledge 4(5):9

Soil Survey Staff. Soil Taxonomy A Basic System of Soil Classification for Making and Interpreting Soil Surveys. Agricultural Handbook No. 436. U.S. Government Printing Office Washington, DC, 1999.

See Also

drainageSuit, erodFUN, erosivFUN

Examples

```
library(sp)
permeabilityClass(11)

texture=suitabinput["texture"]
texture$permeability=permeabilityClass(texture$texture)
str(texture$permeability)
spplot(texture["permeability"])
```

PHSuit *A function for assessing pH suitability requirements for certain crops*

Description

This function determines the suitability classes for soil pH requirements for selected agricultural crops

Usage

```
PHSuit(value, crop)
```

Arguments

value	Input soil pH.
crop	The crop of interest for which soil pH suitability class is sought.

Details

The input value can be map or just a numerical entry of soil pH of a saturated paste extract

Value

The output is pH suitability class for the crop. The output is integer value of suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

If the input value is raster map, then the output will also be a raster map of pH suitability for the crop of interest

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, ECSuit, rainSuit

Examples

```
PHSuit(8.4,"cauliflower")
```

---

predUncertain — *Function to develop spatial map of modelling uncertainty using bootstrap approach*

Description

This functions uses bootstrap approach to estimate spatial maps of modelling prediction interval width and standard deviation

Usage

```
predUncertain(indata,fgrid, k, z, model="rf")
```

Arguments

`indata`	one column input spatial dataframe containing the target soil variable or its transformation
`fgrid`	Input grid or raster stack containing predictors set for the target soil variable
`k`	Set limit for number of realizations/simulations for bootstrap algorithm
`z`	Confidence interval level in percent (for example 95)
`model`	The model for predicting target soil variable using the predictors (for example linear)

Details

One-variable input dataframe is prefered or at least the first column should have the target soil variable to predict. It should not contain NAs. The number of realizations k need not be too high because the software multiplies it exponentially and may slow down the computing process if set to a high value. For example k=5 will results into more than 40 realizations created

Value

a two-layer raster stack map of prediction width and standard deviation

Note

The input dataframe and predictors need to have similar coordinate reference system (CRS). In addition, the input dataframe should not have missing entrie (NAs)

Author(s)

Christian Thine Omuto

References

Efron B. 1992. Jackknife-after-bootstrap standard errors and influence functions. Journal of the Royal Statistical Society. Series B (Methodological), 83–127.

See Also

regmodelSuit, pedoTransfer, imageIndices

Examples

```
library(raster)
library(caret)
soil1=soil[,c("OC")]
predictere=suitabinput[c("depthcodes","rain","texture","dem")]

pred_uncert=predUncertain(soil1,predictere,3,90,"rf")
plot(pred_uncert)
```

`rainSuit` *A function for assessing rainfall suitability requirements for certain crops*

Description

This function determines the suitability classes for rainfall requirements of selected agricultural crops

Usage

```
rainSuit(value, crop)
```

Arguments

`value`	Input rainfall amounts in mm.
`crop`	The crop of interest for which rainfall suitability class is sought.

Details

The input value can be map or just numerical entry of annual rainfall amount in mm

Value

The output is rainfall suitability class for the crop. The output is an integer for suitability class: 1-highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

This function assumes rainfall as the source of water for crop development

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, ESPSuit, classCode

Examples

```
library(sp)
rain=(suitabinput["rain"])
rain$rainmiaz=rainSuit(rain$rain,"wheat")
summary(rain$rainmiaz)
spplot(rain["rainmiaz"])
```

---

reclassifyMap *A function for re-classifying raster maps based on input look-up table*

Description

This function re-classifies an input raster maps based on input look-up table that specifies transition from map classes (or range of classes) to a new class (or range of classes)

Usage

```
reclassifyMap(fgrid,df)
```

Arguments

`fgrid`	Input raster map to be reclassified
`df`	Input look-up table for re-classification

Details

The look-up table should have at least two columns in which the first column contains the classes in the input map and the second column contains the new classes to be assigned

Value

The output is a reclassified raster map

Author(s)

Christian Thine Omuto

References

Robert Hijman. Raster Package in R. https://www.rdocumentation.org/packages/raster

See Also

depthSuit, classCode, rainSuit

Examples

```
library(sp)
LUT=data.frame(map=c(1,2,3,4,5,6),new=c(100,20,30,40,60,80))
newmap=(suitabinput["depthcodes"])
newmap$depth=reclassifyMap(newmap["depthcodes"],LUT)
newmap$melon=depthSuit(newmap$depth,"melon")
summary(newmap$depth)
spplot(newmap["depth"])
```

---

regmodelSuit — *A function for guiding selection of predition model for modelling soil condition*

Description

This function evaluates suitability of most predition models in mapping soil condition using a set of predictors

Usage

```
regmodelSuit(df, ...)
```

Arguments

`df`	a dataframe of target soil property and its predictors
`...`	name of the target soil variable to predict and names of its predictors

Details

The name of the target soil variable to predict and names of its predictors are seperated by commas and are similar to column names of the corresponding variables in the supplied dataframe. The name of the target soil variable starts the list and followed by the names of its predictors. For example, if the dataframe has EC, landcover,DEM, Slope, NDVI, etc., then the input could be (soil,EC,landcover,Slope,DEM).

Value

A table of model statistics such as root mean square error (RMSE), mean absolute error (MAE), r-squared (R2) and Nash-Sutcliffe coefficient of efficiency (NSE) for the popular models in digital soil mapping

Note

The function carries 5-fold cross-validation. Sometimes it may give a warning of missing resample performance with some models. It's important to ensure no NA in the data used for modelling

Author(s)

Christian Thine Omuto

References

Nash, J. E.; Sutcliffe, J. V. 1970. River flow forecasting through conceptual models part I — A discussion of principles. Journal of Hydrology. 10 (3): 282–290

See Also

suitability, pedoTransfer, pedoTransfer, predUncertain

Examples

```
library(caret)
library(sp)
data(soil)
soil1=soil[,c("EC")]
soil1=subset(soil1,!is.na(soil1$EC))
overlay.ov=over(soil1, suitabinput)
soil1$dem=overlay.ov$dem
soil1$rain=overlay.ov$rain
soil1$ph=overlay.ov$ph
soil2=soil1@data[,c("EC","dem","rain","ph")]

regmodelSuit(soil2,EC,dem,rain,ph)
```

RotCmoistcorrection — *A function for estimating moisture effects in RothC carbon turnover modelling*

Description

This function estimates the scalar constant representing the moisture effects in RothC carbon turnover modelling in the soil

Usage

```
RotCmoistcorrection(P, E, S.Thick, clay, pE, fk)
```

Arguments

`P`	the total rainfall amount in mm
`E`	the total evapotranspiration amounts in mm. It can be pan evapotranspiration or potential evapotranspiration rate
`S.Thick`	thickness of soil depth in cm (measured from the soil surface)
`clay`	clay content in percent
`pE`	proportion of pan evapotranspiration representing potential evapotranspiration rate.
`fk`	A constant to correct for soil cover. For bare soil, fk=1.8 and for soil with cover, fk=1

Details

E can be given as pan evapotranspiration or potential evapotranspiration. If potential evapotranspiration is used for E, then pE = 1 and if pan evapotranspiration is used for E then pE=0.75.

Value

A scalar constant for moisture effects on carbon decomposition rates

Note

This function can be used with monthly or annual input data to produce time-dependent scalars

Author(s)

Christian Thine Omuto

References

Burke, I., Kaye, J., Bird, S., Hall, S., McCulley, R., Sommerville, G. 2003. Evaluating and testing models of terrestrial biogeochemistry: the role of temperature in controlling decomposition, Models in ecosystem science, Princeton University Press, Princeton, New Jersey, USA, 225–253, 2003

Adair, E., Parton, W., Del Grosso, S., Silver, W., Harmon, M.,Hall, S., Burke, I., and Hart, S. 2008. Simple three-pool model accurately describes patterns of long-term litter decomposition in diverse climates, Global Change Biology, 14: 2636–2660

Coleman, K. and Jenkinson, D. 2014. ROTHC-26.3 A model for the turnover of carbon in soils: Model description and users guide (Windows version). Rothamsted Research Harpenden Herts AL5 2JQ

See Also

RotCtempcorrection, NPPmodel, carbonTurnover

Examples

```
clay=34.5
depth=30
precip=c(73,59,63,51,52,57,34,55,58,56,76,71)
evapo=c(8,10,27,49,83,99,103,91,69,34,16,8)
inCl=data.frame(seq(1,12,1),precip,evapo)
colnames(inCl)=c("month","rain","ET")
inCl$mcor=RotCmoistcorrection(inCl$rain,inCl$ET,depth,clay,0.75,1)
inCl$mcor
```

`RotCtempcorrection` *A function for estimating temperature effects in organic matter decomposition rates in the soil*

Description

This function estimates the scalar constant for temperature effects in RothC carbon turnover modelling in the soil

Usage

```
RotCtempcorrection(temperature)
```

Arguments

`temperature`	mean air temperature in degrees Celsius

Details

mean air temperature canbe monthly or annual mean temperature

Value

A scalar constant for temperature effects on carbon decomposition rates

Note

This function can be used with monthly or annual input data to produce time-dependent scalars The function works with temperatures greater than -18.2 degrees Celsius

Author(s)

Christian Thine Omuto

References

Burke, I., Kaye, J., Bird, S., Hall, S., McCulley, R., Sommerville, G. 2003. Evaluating and testing models of terrestrial biogeochemistry: the role of temperature in controlling decomposition, Models in ecosystem science, Princeton University Press, Princeton, New Jersey, USA, 225–253, 2003

Adair, E., Parton, W., Del Grosso, S., Silver, W., Harmon, M.,Hall, S., Burke, I., and Hart, S. 2008. Simple three-pool model accurately describes patterns of long-term litter decomposition in diverse climates, Global Change Biology, 14: 2636–2660

Coleman, K. and Jenkinson, D. 2014. ROTHC-26.3 A model for the turnover of carbon in soils: Model description and users guide (Windows version). Rothamsted Research Harpenden Herts AL5 2JQ

See Also

RotCmoistcorrection, NPPmode, carbonTurnover

Examples

```
airTemp=22.1
RotCtempcorrection(airTemp)
```

`saltClass` *A function to classify types of salt-affected soils using EC, PH, and ESP*

Description

This function determines the major classes of salt-affected soils using Electrical Conductivity (EC), soil reaction (pH), and Exchangeable Sodium Percent (ESP) according to FAO or USDA classification schemes

Usage

```
saltClass(ec,ph,ESP,criterion="FAO")
```

Arguments

`ec`	Electrical Conductivity in dS/m of saturated soil paste extract or its equivalent
`ph`	soil reaction (pH)
`ESP`	Exchangeable Sodium Percent
`criterion`	The criterion to use for classifying the soil problem. Either FAO or USDA can be selected

Value

saltClass returns integer classes of salt problems in the soil. The classes are 1, 2, 3, 4, 5 corresponding to None, Saline, Saline-sodic, Sodic, and Alkaline categories.

Note

ESP is mandatory when using this function. The "error: 1 * ESP : non-numeric argument to binary operator" is flagged when ESP entry is missing. In case ESP is missing, saltRating is suggested.

Author(s)

Christian Thine Omuto

References

FAO.2006. Guidelines for soil description. FAO. Rome

Richards, L. A. (ed.) 1954. Diagnosis and Improvement of Saline and Alkali Soils. U.S. Department Agriculture Handbook 60. U.S. Gov. Printing Office, Washington, DC.

See Also

saltRating, saltSeverity, classnames, classCode

Examples

```
library(sp)
saltClass(6.12,7.84, ESP=1,"FAO")

ec=suitabinput["ec"]
ph=suitabinput["ph"]
soc=nutrindicator["soc"]
clay=textureinput["clay"]
texture=suitabinput["texture"]
newmap=ec
newmap$ph=ph$ph
newmap$ECe=ECconversion1(ec$ec,soc$soc,clay$clay,texture$texture,"1:2.5", "FAO")
newmap$salinity=saltClass(newmap$ECe,newmap$ph, ESP=1,"FAO")
newmap$saltclass=classCode(newmap$salinity,"saltclass")
newmap$salineclass1=as.factor(newmap$saltclass)
spplot(newmap["saltclass"], main="Classes of salt-affected soils")
summary(newmap$salineclass1)
```

---

saltRating *A function for classifying salt-affected soils using EC and PH only*

Description

This function determines classes of salt-affected soils using Electrical Conductivity and pH according to FAO or USDA salt classification schemes

Usage

```
saltRating(ec,ph,criterion="FAO")
```

Arguments

ec	Electrical Conductivity in dS/m of saturated soil paste extract or its equivalent
ph	soil reaction (pH)
criterion	The method to use for classifying salt-affected soil. Either FAO or USDA can be selected

Value

The output is an integer value for soil salt class. The class name for any integer code is obtained from classCode function

Note

This function gives approximate classification. A better classification is achieved when indicator of sodium ions is included (e.g. ESP)

Author(s)

Christian Thine Omuto

References

FAO.2006. Guidelines for soil description. FAO. Rome

Richards, L. A. (ed.) 1954. Diagnosis and Improvement of Saline and Alkali Soils. U.S. Department Agriculture Handbook 60. U.S. Gov. Printing Office, Washington, DC.

See Also

saltClass, saltSeverity, classCode

Examples

```
library(sp)
saltRating(11.2,8.14, "USDA")

ec=suitabinput["ec"]
ph=suitabinput["ph"]
soc=nutrindicator["soc"]
clay=textureinput["clay"]
texture=suitabinput["texture"]
newmap=ec
newmap$ph=ph$ph
newmap$ECe=ECconversion1(ec$ec,soc$soc,clay$clay,texture$texture,"1:1", "FAO")
newmap$salinity=saltRating(newmap$ECe,newmap$ph,"FAO")
newmap$salineclass=classCode(newmap$salinity,"saltclass")
newmap$salineclass1=as.factor(newmap$salineclass)
spplot(newmap["salineclass"], main="Soil Salinity Class")
summary(newmap$salinity)
```

saltSeverity — *A function to classify severity levels of salt problems in the soil*

Description

This function classifies degree/severity of salt problems in the soil according to EC, pH and ESP levels

Usage

```
saltSeverity(ec,ph,ESP,method="FAO")
```

Arguments

`ec`	electrical conductivity in dS/m of saturated soil paste extract or its equivalent
`ph`	soil reaction (pH)
`ESP`	Exchangeable sodium percent
`method`	classification method for severity/degree of salt problems. FAO, USDA and Amrhein methods are included. Default method is FAO

Details

This function requires input EC, pH and ESP values to process the classification. They can be maps or numerical entries

Value

Integer classes of ranging between 1-17. The names of integer codes are obtained using classCode function

Note

The function strictly requires input EC, pH, and ESP

Author(s)

Christian Thine Omuto

References

Abrol, IP, Yadav JSP, Massoud FI. 1988. Salt-affected soils and their management. FAO Soils Bulletin 39. FAO, Rome

Amrhein C. 1996. Australian sodic soils: Distribution, properties, and management. Soil Science 161. pp412.

FAO. 2006. Guidelines for soil description. FAO, Rome

Richards LA. 1954. Diagnosis and improvements of saline and alkali soils. Agriculture Handbook No. 60. USDA, Washington

See Also

saltClass, classCode, saltRating

Examples

```
library(sp)
library(rgdal)
saltSeverity(4.5,7.8,11.6,"USDA")
ec=suitabinput["ec"]
ph=suitabinput["ph"]
soc=nutrindicator["soc"]
clay=textureinput["clay"]
texture=suitabinput["texture"]
newmap=ec
newmap$ph=ph$ph
newmap$ECe=ECconversion1(ec$ec*0.1,soc$soc,clay$clay,texture$texture,"1:5", "FAO")
newmap$salt=saltSeverity(newmap$ECe,newmap$ph,0.84,"FAO")
newmap$salineclass=classCode(newmap$salt,"saltseverity")
spplot(newmap["salineclass"], main="Salinity Code")
```

---

slopeSuit — *A function for assessing slope suitability requirements for certain crops*

Description

This function determines the suitability classes for slope requirements of selected agricultural crops

Usage

```
slopeSuit(value, crop)
```

Arguments

`value`	Input land slope in degrees.
`crop`	The crop of interest for which slope suitability class is sought.

Details

The input value can be map or just a numerical entry of slope in degrees

Value

The output is slope suitability class for the crop. The output is an integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

The input slope value must be in degrees

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

LGPSuit, tempSuit, suitability

Examples

```
slopeSuit(23.4,"carrot")
library(sp)
slopeSuit(23.4,"carrot")
slope=suitabinput["slope"]
slope$tea=slopeSuit(slope$slope,"tea")
slope$carrot=slopeSuit(slope$slope,"carrot")
summary(slope$carrot)
spplot(slope["carrot"])
```

---

sloplenFUN — *A function for estimating slope-length factor for soil erosion*

Description

Th function estimates slope length factor for erosion risk assessment. It has options for choosing different algorithms

Usage

```
sloplenFUN(ls,slope,method)
```

Arguments

`ls`	length of slope in metres
`slope`	slope of land in degrees
`method`	method for deriving slope-length factor. The methods included are: WSmith, Renard, Remortel, Zhang, Nearing, Smith, Foster, David, Morgan, and Moore.

Details

Slope (degrees) and length of slope (metres) are relief parameters in erosion risk assessment.

Value

a dimensionless quantity of slope-length factor of erosion risk

Note

The slope must be in degrees. The warning given is a reminder to that the slope is given in degrees

Author(s)

Christian Thine Omuto

References

Benavidez R, Bethana J, Maxwell D, Norton K. 2018. A review of the (Revised) Universal Soil Loss Equation ((R)USLE): with a view to increasing its global applicability and improving soil loss estimates. Hydrol. Earth Syst. Sci., 22, 6059–6086

Omuto CT and Vargas R. 2009. Combining pedometrics, remote sensing and field observations for assessing soil loss in challenging drylands: a case study of northwestern Somalia. Land Degrad. Develop. 20: 101–115

See Also

erosivFUN, erodFUN, slopeSuit

Examples

```
library(sp)
sloplenFUN(60,14.88,"Renard")
newmap=suitabinput["slope"]
newmap$LSrenard=sloplenFUN(60,(newmap$slope),"Renard")
newmap$LSwsmith=sloplenFUN(60,(newmap$slope),"WSmith")
spplot(newmap["LSrenard"])
spplot(newmap["LSwsmith"])
```

SOCSuit — *A function for assessing soil carbon suitability requirements for certain crops*

Description

This function determines the suitability classes for soil organic carbon requirements of selected agricultural crops

Usage

```
SOCSuit(value, crop)
```

Arguments

`value`	Input soil organic carbon content in percent.
`crop`	The crop of interest for which soil organic carbon suitability class is sought.

Details

The input value can be map or just a numerical entry of soil organic carbon in percent

Value

The output is SOC suitability class for the crop. The output is an integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

depthSuit, carbonateSuit, suitability

Examples

```
library(sp)
soc1=nutrindicator["soc"]
soc1$pyrethrum=SOCSuit(soc1$soc,"pyrethrum")
summary(soc1$pyrethrum)
spplot(soc1["pyrethrum"])
```

soil *Sample soil dataset for salinity mapping*

Description

Horizon sample dataset for mapping soil salinity

Usage

```
data("soil")
```

Format

The format is: Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots ..@ data :'data.frame': 152 obs. of 17 variables: .. ..$ Sample : Factor w/ 152 levels "1","10","100",..: 1 65 76 87 98 109 120 131 142 2 ... .$ ProfileID: Factor w/ 87 levels "1","2","3","4",..: 5 53 53 55 55 56 6 7 57 8 ... .$ Latitude : num [1:152] -30.2 -30.3 -30.3 -30.3 -30.3 ... .$ Longitude: num [1:152] 62.2 62.1 62.1 62.1 62.1 ... .$ Horizon : Factor w/ 2 levels "A","B": 1 1 2 1 2 2 1 1 2 1 ... .$ Depth : Factor w/ 43 levels "0 - 100","0 - 17",..: 8 14 37 8 29 42 8 8 38 8 ... .$ Sand : num [1:152] 43.2 61.2 57.2 55.2 65.2 83.2 63.2 63.2 45.2 59.2 ... .$ Silt : num [1:152] 44 24 29 32 22 9 24 24 40 24 ... .$ Clay : num [1:152] 12.8 14.8 13.8 12.8 12.8 7.8 12.8 12.8 14.8 16.8 ... .$ OC : num [1:152] 0.36 0.465 0.39 0.36 0.42 0.87 0.075 0.375 0.84 0.33 ... .$ PH : num [1:152] 8.6 8.37 8.31 8.76 7.81 ... .$ EC : num [1:152] 0.8 2.58 0.98 0.532 1.87 18.5 0.43 0.302 0.345 2.7 ... .$ CaCo3 : num [1:152] 15.2 18.5 20.5 15.8 20 ... .$ K : num [1:152] 67 162 120 124 177 91 127 72 123 158 ... .$ Na : num [1:152] 1073 707 689 646 691 ... .$ CEC : num [1:152] 6 11 18 9 10.4 6 6.4 16 10 4.9 ... .$ ESP : Factor w/ 22 levels "0","1","10","11",..: 11 19 17 20 20 8 17 20 15 11 ... ..@ coords.nrs : num(0) ..@ coords : num [1:152, 1:2] 420924 418226 418226 415334 415334 ... .- attr(*, "dimnames")=List of 2 .. .. ..$: NULL .. .. ..$: chr [1:2] "coords.x1" "coords.x2" ..@ bbox : num [1:2, 1:2] 386582 3343117 427796 3386711 .. ..- attr(*, "dimnames")=List of 2 .. .. ..$: chr [1:2] "coords.x1" "coords.x2" .. .. ..$: chr [1:2] "min" "max" ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot .. .. ..@ projargs: chr "+proj=utm +zone=41 +datum=WGS84 +units=m +no_defs"

Details

A dataset with 87 points of soil horizons for mapping salinity

Source

Hypothetical dataset for salinity mapping

References

Hypothetical dataset for salinity mapping

Examples

```
data(soil)
str(soil)
```

`stoneSuit` *A function for assessing stoniness suitability requirements for certain crops*

Description

This function determines the suitability classes for stoniness requirements of selected agricultural crops

Usage

```
stoneSuit(value, crop)
```

Arguments

`value`	Input level of stoniness in percent.
`crop`	The crop of interest for which stoniness suitability class is sought.

Details

The input value can be map or just a numerical entry of stoniness in percent

Value

The output is stoniness suitability class for the crop. The output is an integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

Output raster map of stoniness for the crop of interest is given if the input value is raster map

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, PHSuit, rainSuit

Examples

```
stoneSuit(15,"grape")
```

`suitability` *A function to determine soil suitability for agricultural crops*

Description

This function determines soil condition classes (such as suitability, fertility, etc.) given a set of indicators.

Usage

```
suitability(df,data)
```

Arguments

`df`	normalized pairwise decision (nxn) matrix for comparing n soil suitability (condition) factors
`data`	a (nxm) matrix of n suitability (condition) factors for m locations (pixels)

Value

A vector of soil suitability (condition) class between 0 and 5.

Note

It's important to normalize and assess the adequacy of the decision matrix before using this function

Author(s)

Christian Thine Omuto

References

FAO, 1976. A framework for land evaluation. FAO Soils Bulletin 32

Saaty TL. 1980. The Analytic Hierarchy Process. McGraw-Hill, New York

See Also

fertilityRating, suitabilityClass

Examples

```
library(sp)
newmap=(nutrindicator)
newmap$carbon=fertilityRating((nutrindicator$soc),"carbon")
newmap$nitrogen=fertilityRating((nutrindicator$nitrogen),"nitrogen")
newmap$potassium=fertilityRating((nutrindicator$potassium),"potassium")
newmap$phosphorus=fertilityRating((nutrindicator$phosphorus),"phosphorus")
newmap$iron=fertilityRating((nutrindicator$iron),"iron")
newmap$zinc=fertilityRating((nutrindicator$zinc),"zinc")
newmap$manganese=fertilityRating((nutrindicator$manganese),"manganese")
newmap$copper=fertilityRating((nutrindicator$copper),"copper")
newmap$cec=fertilityRating((nutrindicator$cec),"cec")
newmap$boron=fertilityRating((nutrindicator$boron),"boron")
newmap$sulfur=fertilityRating((nutrindicator$sulfur),"sulfur")
newmap$soc=NULL
newmapT1=newmap@data
valuT=as.matrix(newmapT1)
data("nutrient")
nutriens=comparisonTable(nutrient)

newmapT1$fertility=suitability(nutrient, valuT)
newmap@data$fertility=newmapT1$fertility
newmap$fertilityclass2=classCode(newmap$fertility,"fertility")
spplot(newmap["fertility"])
summary(newmap$fertilityclass2)
```

---

suitabilityClass *A function to determine suitability classes for given indicator values*

Description

This function determines the suitability class to which a given indicator value falls based on the crop requirement

Usage

```
suitabilityClass(value,crop,factor)
```

Arguments

value	Input indicator value.
crop	The crop of interest for which suitability is determined.
factor	The suitability factor for crop requirement. Example factors include: rain, slope, carbonate, EC, ESP, depth, ph, temperature,

Value

The output is rainfall suitability class for the crop. The output is integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

This function assumes rainfall as the source of water for crop development. The input slope value must be in degrees

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

suitability,slopeSuit, tempSuit

Examples

```
library(sp)
library(raster)
suitabilityClass(20.14,"saffron","slope")
slope=suitabinput["slope"]
slope$tea=slopeSuit(slope$slope,"tea")
slope$saffron=suitabilityClass(slope$slope,"saffron","slope")
summary(slope$saffron)
spplot(slope["tea"], main="Slope suitability for tea")
spplot(slope["saffron"], main="Slope suitability for saffron")
```

---

suitabinput *Sample grid stack map of nutrient indicators for crop fertility requirements*

Description

A grid stack map of nine variables for assessing crop suitabilities

Usage

```
data("suitabinput")
```

Format

The format is: Formal class 'SpatialGridDataFrame' [package "sp"] with 4 slots ..@ data :'data.frame': 16900 obs. of 11 variables: .. ..$ cac03 : num [1:16900] 21.8 20.6 21.2 22 22.3 ... .$ ec : num [1:16900] 2.5 2.38 2.15 2.36 2.24 ... .$ depthcodes: num [1:16900] 3 1 3 3 3 3 3 3 1 1 ... .$ rain : num [1:16900] 282 279 260 279 276 ... .$ texture : int [1:16900] 5 5 5 5 5 5 5 5 11 11 ... .$ dem : num [1:16900] 489 489 489 485 487 ... .$ drainage : int [1:16900] 2 5 2 2 2 5 7 5 5 5 ... .$ stones : num [1:16900] 6 9 6 6 6 6 6 9 9 9 ... .$ structure : int [1:16900] 3 8 7 5 5 5 7 5 9 9 ... .$ ph : num [1:16900] 8.76 8.83 8.73 8.71 8.69 ... .$ slope : num [1:16900] 0.969 0.969 0.969 0.969 0.969 ... ..@ grid :Formal class 'GridTopology' [package "sp"] with 3 slots .. .. ..@ cellcentre.offset: Named num [1:2] 383216 3341506 .. ..- attr(*, "names")= chr [1:2] "x" "y" .. .. ..@ cellsize : num [1:2] 357 357 .. .. ..@ cells.dim : int [1:2] 130 130 ..@ bbox : num [1:2, 1:2] 383038 3341327 429478 3387767 .. ..- attr(*, "dimnames")=List of 2 .. .. ..$: chr [1:2] "x" "y" .. .. ..$: chr [1:2] "min" "max" ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot .. .. ..@ projargs: chr "+proj=utm +zone=41 +datum=WGS84 +units=m +no_defs"

Examples

```
data(suitabinput)
summary(suitabinput$depthcodes)
hist(suitabinput$dem)
```

`surveyPoints` *A function to generate georeferenced locations for monitoring soil conditions*

Description

This function uses stratified random sampling to generate georeferenced locations for monitoring soil conditions

Usage

```
surveyPoints(soilmap,scorpan,conditionclass,mapproportion)
```

Arguments

`soilmap`	input classified map of soil condition
`scorpan`	number of scorpan factors that generated teh soil condition map. The range is 1-5
`conditionclass`	reference class in the soil condition map to be monitored. The class code should be in the map
`mapproportion`	Proportion in percent of the reference class in the soil condition map to be monitored.

Details

The number of scorpan factors can be assumed but need to be with respect to the soil forming factors.The maximum possible number of factors is 5 irrespective of number of layers in each factor while the minimum number is 1.The soil condition class is the class code in the map which is to be targeted

Value

A spatial points dataframe with projection similar to the soil condition map projection

Author(s)

Christian Thine Omuto

See Also

featureRep, imageIndices, pedoTransfer, classCode

Examples

```
library(sp)
library(rgdal)
library(raster)
ec=suitabinput["ec"]
ph=suitabinput["ph"]
soc=nutrindicator["soc"]
clay=textureinput["clay"]
texture=suitabinput["texture"]
newmap=ec
newmap$ph=ph$ph
newmap$ECe=ECconversion1(ec$ec*0.1,soc$soc,clay$clay,texture$texture,"1:5", "FAO")
newmap$salt=saltSeverity(newmap$ECe,newmap$ph,0.84,"FAO")
newmap$salineclass=classCode(newmap$salt,"saltseverity")
newmap$salineclass1=as.factor(newmap$salineclass)
spplot(newmap["salineclass"], main="Salinity Code")
summary(newmap$salt)
summary(newmap$salineclass)
salt=raster(newmap["salt"])
salt1=newmap["salt"]
n_points=surveyPoints(salt1,4,11,80)
length(n_points$new)
spplot(salt1, scales=list(draw=TRUE),sp.layout=list("sp.points",n_points,pch=8,col="cyan"))
spplot(salt, scales=list(draw=TRUE),sp.layout=list("sp.points",n_points,pch=8,col="cyan"))
```

`tempSuit` *A function for assessing temperature suitability requirements for certain crops*

Description

This function determines the suitability classes for temperature requirements of selected agricultural crops

Usage

```
tempSuit(value, crop)
```

Arguments

`value`	Input temperature in degrees Celsius.
`crop`	The crop of interest for which temperature suitability class is sought.

Details

The input value can be map or just a numerical entry of temperature in degrees Celsius

Value

The output is temperature suitability class for the crop. The output is integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

carbonSuit, depthSuite, fertilitySuit

Examples

```
tempgrape=tempSuit(23.5,"grape")
summary(tempgrape)
```

`textureinput` *Sample texture dataset for mapping soil texture*

Description

Sample dataset for assessing soil texture

Usage

```
data("textureinput")
```

Format

The format is: Formal class 'SpatialGridDataFrame' [package "sp"] with 4 slots ..@ data :'data.frame': 16900 obs. of 3 variables: .. ..$ sand: num [1:16900] 61.5 59.8 60.6 58.2 59.1 ... .$ clay: num [1:16900] 12.6 13.9 14.1 13.8 13.8 ... .$ silt: num [1:16900] 25 26.9 25.3 28 26.9 ... ..@ grid :Formal class 'GridTopology' [package "sp"] with 3 slots .. .. ..@ cellcentre.offset: Named num [1:2] 383216 3341506 .. ..- attr(*, "names")= chr [1:2] "x" "y" .. .. ..@ cellsize : num [1:2] 357 357 .. .. ..@ cells.dim : int [1:2] 130 130 ..@ bbox : num [1:2, 1:2] 383038 3341327 429478 3387767 .. ..- attr(*, "dimnames")=List of 2 .. .. ..$: chr [1:2] "x" "y" .. .. ..$: chr [1:2] "min" "max" ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot .. .. ..@ projargs: chr "+proj=utm +zone=41 +datum=WGS84 +units=m +no_defs"

Examples

```
data(textureinput)
summary(textureinput)
```

`textureSuit` *A function for assessing texture suitability requirements for certain crops*

Description

This function determines the suitability classes for texture requirements of selected agricultural crops

Usage

```
textureSuit(value, crop)
```

Arguments

`value`	Input textural class code.
`crop`	The crop of interest for which texture suitability class is sought.

Details

The input value can be map or just a numerical entry of textural class code. The textural class code is obtained using classCode("texture")

Value

The output is texture suitability class for the crop. The output is integer value for suitability class: 1- highly suitable; 2 - moderately suitable; 3 - marginally suitable; 4 - currently not suitable; 5 - not suitable

Note

If the input value is raster map, then the output will also be a raster map of texture suitability for the crop of interest

Author(s)

Christian Thine Omuto

References

Sys, C., Van Ranst, E., Debaveye, J. and Beerneaert, F.1993. Land evaluation: Part III: Crop requirements. Development Cooperation, Belgium.

Naidu, L.G.K., Ramamurthy, V., Challa O., Hegde, R. and Krishnan, P. 2006. Manual, Soil-site Suitability Criteria for Major Crops, National Bureau of Soil Survey and Land Use Planning, ICAR, Nagpur, India

FAO Crop Suitability Requirements: http://ecocrop.fao.org/ecocrop/srv/en/home

See Also

tempSuit, PHSuit, rainSuit

Examples

```
library(sp)
textureSuit(4,"mango")
texture=suitabinput["texture"]
texture$mango=textureSuit(texture$texture,"mango")
summary(texture$mango)
spplot(texture["mango"])
```

Index