

# Package ‘soilassessment’

January 13, 2025

**Type** Package

**Title** Soil Health Assessment Models for Assessing Soil Conditions and Suitability

**Version** 0.3.0

**Description** Soil health assessment builds information to improve decision in soil management. It facilitates assessment of soil conditions for crop suitability [such as those given by FAO <<https://www.fao.org/land-water/databases-and-software/crop-information/en/>>], groundwater recharge, fertility, erosion, salinization [<[doi:10.1002/ldr.4211](https://doi.org/10.1002/ldr.4211)>], carbon sequestration, irrigation potential, and status of soil resources.

**License** GPL

**Depends** R (>= 3.5.0)

**Imports** caret, deSolve, e1071, FuzzyAHP, googledrive, Hmisc, methods, nnet, png, randomForest, raster, soiltexture, sp, sf, terra, stats

**Suggests** arm, Cubist, aqp, kernlab, nlme, ranger, rpart, plyr, qrn, quantregForest, knitr, rmarkdown

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Christian Thine Omuto [aut, cre]  
(<<https://orcid.org/0000-0002-5792-3485>>)

**Maintainer** Christian Thine Omuto <[thineomuto@yahoo.com](mailto:thineomuto@yahoo.com)>

**Repository** CRAN

**Date/Publication** 2025-01-13 14:10:01 UTC

## Contents

appendTextureclass . . . . .	3
------------------------------	---

carbonateSuit . . . . .	4
carbonTurnover . . . . .	6
CECSuit . . . . .	7
classCode . . . . .	8
classLUT . . . . .	9
classnames . . . . .	10
comparisonTable . . . . .	11
conversion . . . . .	12
createTexturedata . . . . .	13
DataAvailabilityIndex . . . . .	14
depthharm . . . . .	15
depthSuit . . . . .	16
drainageSuit . . . . .	18
ECconversion1 . . . . .	19
ECconversion2 . . . . .	20
ECconversion3 . . . . .	22
ECconversion4 . . . . .	23
ECharm_Info . . . . .	24
ECSuit . . . . .	25
erodFUN . . . . .	26
erodibilityRisk . . . . .	27
erosivFUN . . . . .	28
ESPSuit . . . . .	29
featureRep . . . . .	31
fertilityRating . . . . .	32
fertilitySuit . . . . .	33
harmonization . . . . .	34
imageIndices . . . . .	35
LGPSuit . . . . .	37
ME_ECharm . . . . .	38
ME_ECharmserve . . . . .	39
ME_PHharm . . . . .	40
ME_PHharmserve . . . . .	41
NDVIcoverfactor . . . . .	41
negData . . . . .	42
NPPmodel . . . . .	43
nutrient . . . . .	44
nutrindicator . . . . .	45
pedoTransfer . . . . .	46
permeabilityClass . . . . .	47
PHConversion . . . . .	48
PHharm_Info . . . . .	50
PHSuit . . . . .	51
predAccuracy . . . . .	52
predUncertain . . . . .	53
rainSuit . . . . .	54
reclassifyMap . . . . .	55
regmodelSuit . . . . .	56

RotCmoistcorrection . . . . .	58
RotCtempcorrection . . . . .	59
saltClass . . . . .	60
saltRating . . . . .	61
saltSeverity . . . . .	63
SASdata_densityInfo . . . . .	64
SASglobeData . . . . .	65
SASmodels . . . . .	66
slopeSuit . . . . .	67
sloplenFUN . . . . .	69
SOCSuit . . . . .	70
soil . . . . .	71
stoneSuit . . . . .	72
suitability . . . . .	73
suitabilityClass . . . . .	74
suitabinput . . . . .	76
surveyPoints . . . . .	76
tempSuit . . . . .	78
textureinput . . . . .	79
textureSuit . . . . .	79
VegCOV . . . . .	81

<b>Index</b>	<b>83</b>
--------------	-----------

---

appendTextureclass	<i>A function for attaching soil textural classes</i>
--------------------	---

---

## Description

This function attaches soil textural classes according to 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. Example 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](https://cran.r-project.org/web/packages/soiltexture/vignettes/soiltexture_vignette.pdf)

**See Also**

[textureSuit](#), [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 and trees*

---

**Description**

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

**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 Beerneart, 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](#), [ESPSuit](#), [fertilitySuit](#)

### Examples

```
library(sp)
newmap=suitabinput
newmap$saffron=carbonateSuit(newmap$cac03,"saffron")
summary(newmap$saffron)
splot(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 ordinary 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**

[RotCmoistcorrection](#), [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 and trees*

---

**Description**

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

**Usage**

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

**Arguments**

value           input Cation Exchange Capacity (Cmol(+)/kg).  
crop             crop of interest for which CEC suitability class is sought

**Details**

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 Beerneart, 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](#), [tempSuit](#), [rainSuit](#)

**Examples**

```
CECSuit(22.4, "pineapple")
```

---

classCode	<i>A function for displaying names of class codes of soil conditions in the soilassessment package</i>
-----------	--

---

**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	<i>A function for developing Look-up Table (LUT) for the soil condition class map</i>
----------	---

---

**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	<i>A function to display class names and codes as used in the soilassessment package</i>
------------	--

---

**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	<i>A function for normalizing decision ranking table</i>
-----------------	--

---

**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](#), [fertilitySuit](#)

**Examples**

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

---

conversion

*A function for developing own harmonization model*

---

### Description

This function enables development of own function for harmonizing soil indicators to standard values

### Usage

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

### Arguments

x	input predictor value
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](#), [ECconversion4](#)

**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~conversion(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)
```

---

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 per cent

**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**

[createTexturedata](#), [appendTextureclass](#)

**Examples**

```
#data(textureinput)
newmap=textureinput

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

---

DataAvailabilityIndex *A function for showing sampling point density map in a geographic area*

---

**Description**

An index map of density of sampling points in a geographic area

**Usage**

```
DataAvailabilityIndex(Boundary, Scale, CP, Data)
```

**Arguments**

Boundary	a spatial polygon or data frame of coordinates of vertices of a bounding geographic area where data search is intended
Scale	unit area to show spatial density of available sampled points
CP	coordinate projection of the Boundary spatial polygon
Data	input spatial spreadsheet database containing all possible point samples

**Details**

The input spreadsheet database should contain spatial coordinates of available samples. Example input spatial spreadsheet database is the global soil database. The Scale should be provided in area units e.g., 0.5, 1, 20, 30 (square km). Large areas cover more data than small areas. Hence, they take time to process. Coordinate projection (CP) for Boundary polygon should be of formal class crs (coordinate reference system). It's preferable to provide CP for Boundary area similar to CP for input data

**Value**

A spatial raster map of density of sample locations per unit (specified) area

**Note**

Scales less than 0.1 square km may be too small for search. Large scales (say 10000 square km) may be too large and take time to process

**Author(s)**

Christian Thine Omuto

**Examples**

```
library(sp)
library(raster)
library(terra)
library(sf)
x <- c(20.02,25.69,25.69,20.02)
y <- c(-28.40,-32.76,-32.76,-34.84)
yx=data.frame(cbind(x, y))
CRs="+proj=longlat +datum=WGS84 +no_defs"
Data=SASglobeData("ph","ZAF")
coordinates(Data)=~Longitude+Latitude
crs(Data)=CRs
Index=DataAvailabilityIndex(yx,60,CRs,Data)
plot(Index)
```

---

depthharm

*A function for harmonizing soil property between uniform depth intervals in observation pits*

---

**Description**

A function to harmonize soil property between uniform depth intervals in a set of observation pits

**Usage**

```
depthharm(soildata, var.name, lam, d)
```

**Arguments**

soildata	soil data containing soil property to be harmonized and observed depth intervals
var.name	name of target variable or soil property to be harmonized
lam	a factor to improve prediction of target soil property between sampled depths
d	target uniform depth intervals for harmonizing the target soil property

**Details**

Input soil data must be a dataframe or class of ProfileCollection. The smoothing factor improves prediction of the target soil property. Its default value is 0.1. Desired depth intervals are separated by comma and should be chosen between minimum and maximum depths in the soil data.

**Value**

The output is a list of two dataframes: `harmonized.d` is a dataframe of harmonized soil property at target depth intervals. `obs_n_pred` is augmented dataframe of observed and harmonized soil properties

**References**

Bishop, T.F.A., McBratney, A.B., Laslett, G.M., 1999. Modelling soil attribute depth functions with equal-area quadratic smoothing splines. *Geoderma* 91, 27–45. [https://doi.org/10.1016/S0016-7061\(99\)00003-8](https://doi.org/10.1016/S0016-7061(99)00003-8)

Malone, B.P., McBratney, A.B., Minasny, B., Laslett, G.M., 2009. Mapping continuous depth functions of soil carbon storage and available water capacity. *Geoderma* 154, 138–152. <https://doi.org/10.1016/j.geoderma.2009.10.011>

Ponce-Hernandez, R., Marriott, F.H.C., Beckett, P.H.T., 1986. An improved method for reconstructing a soil profile from analyses of a small number of samples. *Journal of Soil Science* 37, 455–467. <https://doi.org/10.1111/j.1365-2389.1986.tb00377.x>

**Examples**

```
library(aqp)
library(plyr)
library(sp)
x=c(rep(2.12,4),rep(2.05,4))
y=c(rep(9.34,4),rep(8.17,4))
pit=c(rep(1,4),rep(2,4))
depthcode=c(1,2,3,4,1,2,3,4)
upper=c(0,18,25,35,0,12,33,50)
lower=c(10,25,35,67,12,33,50,100)
pH=c(6.7,5.5,5.1,6.7,6.4,5.8,5.3,5.0)
df=data.frame(pit,x,y,upper,lower,depthcode,pH)
lat=df$x;lon=df$y;id=df$pit;top=df$upper;
bottom=df$lower;horizon=df$depthcode;Varn=df$pH

soildata <- join(data.frame(id, top, bottom, Varn, horizon),
                data.frame(id, lat, lon), type='inner')
depths(soildata) <- id ~ top + bottom
site(soildata) <- ~ lat + lon
initSpatial(soildata, crs = "EPSG:4326") <- ~ lat + lon
depth.s = depthharm(soildata, var.name= "Varn",
                    lam=0.01,d = t(c(0,10,40,80,100,150)))
plot(soildata, color= "Varn", name="horizon")
```

---

depthSuit

*A function for assessing soil depth suitability requirements for certain crops and trees*

---

**Description**

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



**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 Beerneart, 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](#), [tempSuit](#), [fertilitySuit](#)

**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)
splot(newmap["depth"])
```

---

drainageSuit	<i>A function for assessing drainage suitability requirements for certain crops and trees</i>
--------------	---

---

**Description**

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

**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 Beerneart, 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](#), [PHSuit](#), [rainSuit](#)

**Examples**

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

---

ECconversion1	<i>A function for harmonizing electrical conductivity of a soil solution to that of the saturated paste extract</i>
---------------	---

---

**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,texture,method,extract,oc=NULL,clay=NULL)
```

**Arguments**

ec	measured electrical conductivity of the soil solution
texture	soil textural class according to USDA or its equivalent. Texture class is given in terms of class codes as given in <code>classnames("texture")</code>
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
extract	ratio of soil:water in extract solution for measuring electrical conductivity was measured. Example is 1:1, 1:2, etc. The default is 1:1
oc	organic matter content of the soil in percent
clay	clay content of the soil in percent

**Details**

This function considers the influence of texture and soil-water solution on conversion of electrical conductivities. The functions includes FAO, sonmez, and hogg conversion models. FAO model requires information on clay content and organic carbon content.

**Value**

equivalent electrical conductivity of saturated soil extract

**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](#), [ECconversion3](#), [ECconversion4](#)

**Examples**

```
library(sp)
library(raster)
ECconversion1(7.31,"SiCl","FA0","1:2.5",0.91,22.5)
ec=suitabinput["ec"]
soc=nutrindicator["soc"]
clay=textureinput["clay"]
texture=suitabinput["texture"]
newmap=ec
newmap$ECe=ECconversion1(ec$ec,texture$texture,"FA0","1:2.5",soc$soc,clay$clay)
splot(newmap["ECe"], main="Equivalent ECse")
```

---

ECconversion2

*A function for harmonizing electrical conductivity of a soil solution to that of the 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, method, extract)
```

**Arguments**

ec	measured electrical conductivity of the soil solution in dS/m
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, hogg, park, visconti, korsandi, shahid, klaustermeier, and he. The default is USDA

extract            ratio of soil:water in extract solution for measuring electrical conductivity was measured. Example is 1:1, 1:2, etc. Default is 1:1

### 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

### Note

Models that work with soil solutions in 1:1 soil-water mix are: USDA, landon, kargas, ozkan,hogg, and zhang. Models for 1:2 solutions are: USDA and hogg. Models for 1:2.5 are: ozkan and shahid. landon model also works for 1:3 soil solution. Models for 1:5 are: USDA, landon, kargas, ozkan, chi, park, visconti, korsandi, klaustermeier, and he. The function only works for soil solution mix ratio handled by the respective model.

### 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](#), [ECconversion3](#), [ECconversion4](#)

### Examples

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

---

`ECconversion3`*A function for developing own harmonization model*

---

**Description**

This function enables development of own function for harmonizing soil indicators to standard values

**Usage**

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

**Arguments**

<code>x</code>	input predictor value
<code>A</code>	location parameter representing the value of target variable when the predictors are minimal (or the y-intercept)
<code>B</code>	Rate parameter representing the rate of change of the target variable with the predictor (or the slope)
<code>method</code>	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](#), [ECconversion4](#)

**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 harmonizing 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.

**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](#), [pedoTransfer](#)

**Examples**

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

---

ECharm_Info	<i>Information on performance of soil electrical conductivity (EC) harmonization models</i>
-------------	---

---

**Description**

Information index for relative predictive performance of soil EC harmonization models

**Usage**

```
ECharm_Info(solution)
```

**Arguments**

solution	ratio of soil-water solution for the extract used in measuring electrical conductivity.
----------	---

**Details**

Ratio in text format for the soil-water solution for the extract used in measuring EC. It's given in quotation marks. Current models consider "1:2", "1:2.5", and "1:5" ratios. Default ratio is "1:2"

**Value**

Graphical display of the predictive performance index for the harmonization models in different regions of the world: Africa, Asia, Near East and North Africa (NENA), Latin America and Caribbean (LAC), north America, and Europe. The performance index ranges between 0 (poor) to 1 (best).

**Note**

The function currently works for 1:2, 1:2.5, and 1:5. These ratios must be entered in quotation marks. Due to periodic update, internet connectivity is needed for the function to work.

**Author(s)**

Christian Thine Omuto

**See Also**

[PHharm\\_Info](#), [SASdata\\_densityInfo](#)



**Examples**

```
ECharm_Info("1:2")
```

---

ECSuit	<i>A function for assessing Electrical Conductivity suitability requirements for certain crops and trees</i>
--------	--

---

**Description**

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

**Usage**

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

**Arguments**

value	Input electrical conductivity in dS/m.
crop	The crop of interest for which EC 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**

Should the input value be 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 Beerneart, 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](#), [PHSuit](#), [fertilitySuit](#)

**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,texture$texture,"FA0","1:2.5",soc$soc,clay$clay)
newmap$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 <code>classnames("texture")</code> to insert the correct texture code
Struct	soil structure code representing the soil structure class. Use <code>classnames("structure")</code> 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](#), [sloplenFUN](#)

**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)
splot(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](#), [sloplenFUN](#)

**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)
splot(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 and trees*

---

**Description**

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

**Usage**

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

**Arguments**

value	Input Exchangeable Sodium Percent (ESP)
crop	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 Beerneart, 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](#), [rainSuit](#), [fertilitySuit](#)

**Examples**

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

---

featureRep	<i>A function to assess how well landscape features are represented in discrete samples</i>
------------	---

---

### 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](#)

**Examples**

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

---

fertilityRating	<i>A function for determining soil fertility levels for given soil property (fertility indicator)</i>
-----------------	---

---

**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**

[suitability](#), [saltRating](#), [fertilitySuit](#)

**Examples**

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

---

fertilitySuit	<i>A function for assessing soil fertility suitability requirements for certain crops</i>
---------------	---

---

**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 Beerneart, 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](#), [ESPSuit](#), [fertilityRating](#)

## 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)
splot(newmap["fertilityokra"], main="Fertility suitability map for Okra")
```

## Description

A generic model for harmonizing soil data for salt-affected soils.

**Usage**

```
harmonization(x,A,B)
```

**Arguments**

x - is input data to harmonize such as electrical conductivity (ec) or ph.  
A - is real number rate parameter or slope of the harmonization model  
B - is real number constant (intercept) of the harmonization model

**Details**

This is a generic linear model for harmonizing input soil data for assessing salt-affected soils.

**Value**

a numeric output of harmonized ec or ph

**Author(s)**

Christian Thine Omtó

**See Also**

[ME\\_ECharmserve](#), [ME\\_PHharmserve](#)

**Examples**

```
A = 1.08  
B = 0.303  
ec=2.45  
harmonization(2.45,1.08,0.303)
```

---

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 $\mu\text{m}$
green	green image band with wavelength range: 0.533-0.59 $\mu\text{m}$
red	red image band with wavelength range: 0.636-0.673 $\mu\text{m}$
nir	NIR image band with wavelength range: 0.851-0.879 $\mu\text{m}$
swir1	SWIR image band with wavelength range: 1.566-1.651 $\mu\text{m}$
swir2	SWIR image band with wavelength range: 2.107-2.294 $\mu\text{m}$
index	index from combination of image bands such as NDVI, SAVI, SI, etc. The default is NDVI.

**Details**

The indices are based on multispectral bands: blue, green, red, NIR(near infrared), SWIR1 (short-wave infrared1) and SWIR2(short-wave infrared2)

**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	<i>A function for assessing Length of Growing Period (LGP) suitability requirements for certain crops and trees</i>
---------	---

---

**Description**

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

**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 Beerneart, 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](#), [PHSuit](#), [fertilitySuit](#)

**Examples**

```
library(sp)
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")
```

---

ME\_ECharm

*Mixed-effects model for harmonizing soil electrical conductivity to the equivalent conductivity of saturated paste extract*

---

**Description**

A function for harmonizing soil electrical conductivity to the equivalent conductivity of saturated paste extract using mixed effects approach

**Usage**

```
ME_ECharm(EC, TEXCLASS, model, soilsolution)
```

**Arguments**

EC	a vector or single value of soil electrical conductivity to be harmonized. It should have been determined in a given soil solution (e.g. 1:2, 1:2.5 or 1:5)
TEXCLASS	soil textural class of the soil whose electrical conductivity is to be harmonized. String or test entry of USDA textural classes: Cl, ClLo,Lo,LoSa,Sa,SaCl,SaClLo,SaLo,SiCl,SiClLo,SiLo, The classes can be determined from Clay, Silt, and Sand proportions using createTexturedata function
model	functional model for relating EC to be harmonized and equivalent EC of saturated paste extract. Models considered are second order polynomial, sigmoid, spherical, gaussian, exponential, power, and linear functions. The default is polynomial
soilsolution	– soil:water mix ratio in which electrical conductivity was measured. The function is currently working on 1:2, 1:2.5, and 1:5. The default is 1:2

**Details**

EC harmonization models, which were developed using global datasets, are designed to standardize soil electrical conductivity for applications in soil salt classification

**Value**

numeric value of equivalent EC of saturated soil paste extract

**Note**

The models are currently developed for soil solutions from 1:2, 1:2.5 and 1:5 soil:water mix ratios. The function only works with USDA soil textural classes. Convert other soil textural classes to USDA classes for all applications with this function.

**Author(s)**

Christian Thine Omuto

**References**

Omuto, C. T., Vargas, R.-R., EL Mobarak, A., Nuha, M., Viatkin, K., & Yigini, Y. (2020). Mapping of salt-affected soils – Technical manual. FAO. <https://doi.org/10.4060/ca9215en>

Omuto, C. T., Minasny, B., McBratney, A. B., & Biamah, E. K. (2006). Nonlinear mixed effect modelling for improved estimation of water retention and infiltration parameters. *Journal of Hydrology*, 330(3–4), 748–758. <https://doi.org/10.1016/j.jhydrol.2006.05.006>

Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-Effects Models in Sand S-PLUS*. Springer New York. <https://doi.org/10.1007/978-1-4419-0318-1>

**See Also**

[ME\\_PHharm](#), [ECconversion1](#), [ECconversion2](#)

**Examples**

```
ndata=data.frame(EC=c(1,0.34,5.07,12.17, 2.219),TEX=c("Cl","SaCl","LoSa", "SiCl","SaClLo"))
ndata$ESa1=ME_ECharm(ndata$EC,ndata$TEX,"power", "1:5")
```

---

ME\_ECharmserve

*Harmonization models for soil electrical conductivity*

---

**Description**

Mixed effects models for harmonizing electrical conductivity (EC)

**Details**

Suit of mixed-effects models

**Note**

Internet connectivity is needed for the function to work.

---

ME_PHharm	<i>Mixed-effects model for harmonizing soil pH (KCl or CaCl<sub>2</sub>) to the equivalent pH (water)</i>
-----------	---

---

**Description**

A function for harmonizing soil pH (KCl or CaCl<sub>2</sub>) to the equivalent pH (water) using mixed effects approach

**Usage**

```
ME_PHharm(ph, TEXCLASS, model, phtype)
```

**Arguments**

ph	a vector or single value of soil ph in KCl or CaCl <sub>2</sub> to be harmonized
TEXCLASS	soil textural class of the soil whose ph is to be harmonized. String or test entry of USDA textural classes: Cl, ClLo, Lo,LoSa,Sa,SaCl,SaClLo,SaLo,SiCl,SiClLo,SiLo,Si,CS,MS,HCL,FS. The classes can be determined from Clay, Silt, and Sand proportions using createTexturedata function
model	functional model for relating ph in KCl or CaCl to be harmonized and equivalent ph (water). Models considered are second order polynomial, sigmoid, spherical, gaussian, exponential, power, and linear functions. The default is polynomial
phtype	KCl or CaCl <sub>2</sub> solution for ph. The default is CaCl <sub>2</sub>

**Details**

ph harmonization models, which were developed using global datasets, are designed to standardize soil ph for applications in soil salt classification

**Value**

numeric value of equivalent ph (water)

**Note**

The function only works with USDA soil textural classes. Convert other soil textural classes to USDA classes for all applications with this function.

**Author(s)**

Christian Thine Omuto

**See Also**

[ME\\_ECharm](#), [ECconversion1](#), [ECconversion2](#)



**Examples**

```
newdata=data.frame(ph=c(1.6,8.3,5.7,12.1,2.2),tex=c("Cl","SaCl","LoSa","Si","SaClLo"))
newdata$pH2=ME_PHharm(newdata$ph,newdata$tex,"exponential","kcl")
```

---

ME_PHharmserve	<i>Harmonization models for soil pH</i>
----------------	---

---

**Description**

Mixed effects models for harmonizing soil pH

**Details**

Suit of mixed-effects models

**Note**

Internet connectivity is needed for the function to work.

---

NDVICoverfactor	<i>Estimating Cover Factor for Erosion Models Using NDVI Remote Sensing Index for Vegetation</i>
-----------------	--

---

**Description**

This function estimates the C factor of erosion models using NDVI derived from remote sensing images

**Usage**

```
NDVICoverfactor(ndvi, model="kniff")
```

**Arguments**

ndvi	remote sensing index for vegetation
model	model for relating ndvi with C factor

**Details**

The function uses empirical models from the literature. It currently has 15 models: kniff, patil, almagro, jamshidi, dejong, toumi, gitas, joshi, durigon, wickama, suriyaprasit, lin, bahrawi, kulikov, power, modis.

**Value**

a numeric value between 0 and 1 .

**Note**

Some models may return negative values, which are masked out in the function or may return warning

**Author(s)**

Christian Thine Omuto

**References**

Ayalew, D.A., Deumlich, D., Šarapatka, B., Doktor, D., 2020. Quantifying the Sensitivity of NDVI-Based C Factor Estimation and Potential Soil Erosion Prediction using Spaceborne Earth Observation Data. *Remote Sensing* 12, 1136. <https://doi.org/10.3390/rs12071136>

Mahgoub, M., Elalfy, E., Soussa, H., Abdelmonem, Y., 2024. Relation between the soil erosion cover management factor and vegetation index in semi-arid basins. *Environ Earth Sci* 83, 337. <https://doi.org/10.1007/s12665-024-11593-3>

**See Also**

[VegCOV](#)

**Examples**

```
NDVICoverfactor(0.27, "modis")

library(raster)
library(sp)
r <- raster(xmn= 35.5, ymn= -1.5, xmx = 37.5,ymx = 1.5, res = c(0.01,0.01),
            crs = '+proj=latlon +datum=WGS84 +no_defs')
r <- setValues(r, sample(x=0:1, size=ncell(r), replace=TRUE))
r$s2=NDVICoverfactor(values(r),"modis")
plot(r$s2)
```

---

negData

*Correcting negative entries in classes for intenisty of salt-affected soils*

---

**Description**

Function to handle negative entries when assessing salt-affected soils

**Usage**

```
negData(vg, x)
```

**Arguments**

vg	tag for soil property. Default is "ec"
x	numeric value of soil property to check

**Details**

Three tags for soil properties are allowed: "ec", "ph", "esp"

**Value**

numeric value of soil property to correct. It return NA where negative "ec" or "esp" is involved or where  $ph < 1$  or  $ph > 14$

**Author(s)**

Christian Thine Omuto

**Examples**

```
negData("ph", 14)
```

---

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/m<sup>2</sup>/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**

[carbonTurnover](#), [RotCmoistcorrection](#)

**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 reciprocals 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 .. ..@ 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)
#splot(nutrindicator["nitrogen"])
```

---

pedoTransfer	<i>A pedotransfer function to predict electrical conductivity or any other soil property using other soil properties</i>
--------------	--

---

### 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 properties
...	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

[ECconversion4](#), [conversion](#)

### 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                  is a string describing soil textural class

### Details

Soil textural class is according to USDA textural triangle such as SiLo, Si, SaLo. The code is represented by first two letters of the class with the first letter in upper case, e.g., Si, Lo, Cl. Class codes with combination of textures have the first two letters included in the combination, e.g., SiLo, SaCilo, etc.

**Value**

permeability class code. 1-very slow, 2-slow, 3-moderately slow, 4-moderate, 5-moderately rapid, 6-rapid, and 7-very rapid

**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](#)

**Examples**

```
library(sp)
permeabilityClass("SaLo")

texture=suitabinput["texture1"]
texture$permeability=ifelse(texture$texture1=="Lo",
permeabilityClass("Lo"),ifelse(texture$texture1=="SaLo",
permeabilityClass("SaLo"),permeabilityClass("SiLo")))
str(texture$permeability)
texture$Perm=classCode(texture$permeability,"permeability")
splot(texture["Perm"])
```

---

PHConversion	<i>Models for converting soil pH (KCl or CaCl<sub>2</sub>) to the equivalent pH (water)</i>
--------------	---

---

**Description**

A suit of functions for converting soil pH (KCl or CaCl<sub>2</sub>) to the equivalent pH (water)

**Usage**

```
PHConversion(ph, model,phtype)
```



**Arguments**

ph	a vector or single value of soil ph in KCl or CaCl <sub>2</sub> to be converted to ph (water)
model	functional model for relating ph in KCl or CaCl to be converting and equivalent ph (water). Models considered are second order kabala, sadovski, davies, brennan functions. The default is kabala
phtype	KCl or CaCl <sub>2</sub> solution for ph. The default is CaCl <sub>2</sub>

**Details**

ph conversion models are those in the literature

**Value**

numeric value of equivalent ph (water)

**Note**

ph ranges between 1 and 14

**Author(s)**

Christian Thine Omuto

**References**

- Davies, B.E. (1971). A Statistical Comparison of pH Values of some English Soils after Measurement in both Water and 0.01M Calcium Chloride. *Soil Science Society of America Journal* 35, 551–552. <https://doi.org/10.2136/sssaj1971.03615995003500040022x>
- Kabała, C., Muszyfaga, E., Gałka, B., Łabuńska, D., Mańczyńska, P. (2016). Conversion of Soil pH 1:2.5 KCl and 1:2.5 H<sub>2</sub>O to 1:5 H<sub>2</sub>O: Conclusions for Soil Management, Environmental Monitoring, and International Soil Databases. *Pol. J. Environ. Stud.* 25, 647–653. <https://doi.org/10.15244/pjoes/61549>
- Miller, R.O., Kissel, D.E. (2010). Comparison of Soil pH Methods on Soils of North America. *Soil Sci. Soc. Am. J.* 74, 310–316. <https://doi.org/10.2136/sssaj2008.0047>
- Sadovski, A.N. (2019). Study on pH in water and potassium chloride for Bulgarian soils. *EURASIAN JOURNAL OF SOIL SCIENCE (EJSS)* 8, 11–16. <https://doi.org/10.18393/ejss.477560>

**See Also**

ME\_PHharm, ME\_ECharm, ECconversion1

**Examples**

```
testdata=data.frame(PHKC=c(6.45,8.34,5.07,12.17, 4.219),TEX=c("Cl","SaCl","LoSa","Si","SaClLo"))
testdata$PHs1=PHConversion(testdata$PHKC,"kabala","kcl")
```

---

PHharm\_Info

*Information on performance of soil pH (water) harmonization models*

---

### **Description**

Information index for relative predictive performance of soil pH harmonization models

### **Usage**

```
PHharm_Info(solution)
```

### **Arguments**

`solution`            solution for measuring soil pH

### **Details**

Solution for measuring soil pH. It's given in quotation marks. Current models consider "cac12" and "kcl" ratios. Default solution is "cac12"

### **Value**

Graphical display of the predictive performance index for the harmonization models in different regions of the world: Africa, Asia, Near East and North Africa (NENA), Latin America and Caribbean (LAC), north America, and Europe. The performance index ranges between 0 (poor) to 1 (best).

### **Note**

The function currently works for cac12 and kcl. These solutions must be entered in quotation marks. Due to periodic update, internet connectivity is needed for the function to work.

### **Author(s)**

Christian Thine Omuto

### **See Also**

[ECharm\\_Info](#), [SASdata\\_densityInfo](#)

### **Examples**

```
PHharm_Info("kcl")
```

---

PHSuit	<i>A function for assessing pH suitability requirements for certain crops and trees</i>
--------	---

---

**Description**

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

**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 Beerneart, 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](#), [ECSuit](#), [fertilitySuit](#)

**Examples**

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

---

predAccuracy

*A function for accuracy assessment between an array of two variables*

---

**Description**

This function calculates statistical indices for accuracy between an array of two variables such as calibration and validation vectors. The indices are Bias, RMSE, R-squared, and NSE

**Usage**

```
predAccuracy(x, y)
```

**Arguments**

x                    a numeric vector of first variable of the two variables for accuracy assessment  
y                    a numeric vector of second variable of the two variables for accuracy assessment

**Details**

The function calculates four indices for accuracy: bias, root mean square error (RMSE), r-squared, Nash-sutcliffe efficiency (NSE)

**Value**

A table of four variables: bias, RMSE, Rsquared, and NSE

**Author(s)**

Christian Thine Omuto

**References**

Becker, R. A., Chambers, J. M. and Wilks, A. R. 1988. The New S Language. Wadsworth & Brooks/Cole.

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**

[predUncertain](#), [featureRep](#)

**Examples**

```
xy=data.frame(a=c(2,3,4,5,6,7,8,9),b=c(1,1.5,8,10,12,3.5,NA,18))
predAccuracy(xy$a,xy$b)$Rsquared
```

---

predUncertain	<i>A function to develop spatial map of modelling uncertainty using the bootstrap approach</i>
---------------	--

---

**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 preferred 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](#), [imageIndices](#), [predAccuracy](#)

**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	<i>A function for assessing rainfall suitability requirements for certain crops and trees</i>
----------	---

---

**Description**

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

**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 Beerneart, 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](#), [ESPSuit](#), [fertilitySuit](#)

**Examples**

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

[classCode](#), [classLUT](#), [classnames](#)

**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)
splot(newmap["depth"])
```

---

regmodelSuit

*A function for guiding selection of a prediction model for modelling soil properties*

---

**Description**

This function evaluates suitability of most prediction models in mapping soil properties 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 separated 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

[pedoTransfer](#), [predUncertain](#), [ECconversion3](#)

## 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

[carbonTurnover](#), [RotCtempcorrection](#), [NPPmodel](#)

## 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	<i>A function for estimating temperature effects in organic matter decomposition rates in the soil</i>
--------------------	--

---

## 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 can be 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**

[carbonTurnover](#), [RotCmoistcorrection](#), [NPPmodel](#)

**Examples**

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

---

saltClass	<i>A function to classify types of salt-affected soils using EC, PH, and ESP</i>
-----------	--

---

**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)
```

**Arguments**

ec	Electrical Conductivity in dS/m of saturated soil paste extract or its equivalent
ph	soil reaction (pH)
esp	Exchangeable Sodium Percent

**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](#), [classCode](#)

**Examples**

```
saltClass(6.12,7.84,1)
```

---

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,texture$texture,"FAO","1:1", soc$soc,clay$clay)
newmap$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	<i>A function to classify salt intensity in soil</i>
--------------	--

---

### Description

This function classifies salt intensity in soil based on EC, pH and ESP levels

### Usage

```
saltSeverity(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	classification method for severity/degree of salt problems. FAO, USDA, Amrhein, and PSALT criteria 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. PSALT criterion uses percent salt content instead of EC.

### 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. Percent salt content can be used in place of EC if the criterion is PSALT

### 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](#), [saltRating](#), [classCode](#)

**Examples**

```
library(sp)
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-1
newmap$ECe=ECconversion1(ec$ec*0.25,texture$texture,"FA0","1:5",soc$soc,clay$clay)
newmap$salt=saltSeverity(newmap$ECe,newmap$ph,6.84,"FA0")
newmap$salineclass=classCode(newmap$salt,"saltseverity")
splot(newmap["salineclass"], main="Salinity Code")
```

---

SASdata_densityInfo	<i>Information on global spatial distribution of locations with measured soil properties for salt-affected soils (SAS)</i>
---------------------	--

---

**Description**

Global distribution of sampling points with measured soil property data

**Usage**

```
SASdata_densityInfo(data)
```

**Arguments**

data	type of measured soil data in the global database of SAS information. There are three categories of soil data: ec, ph, texture.
------	---

**Details**

The function accepts three input alternatives for querying available SAs information: "ec", "ph", and "texture". The default is "ec"

**Value**

Spatial maps of sampling locations with measured soil data for SAS information. They include maps of electrical conductivity (ec), pH, texture (sand, silt, clay percentages), and cation exchange capacity (CEC). Locations for CEC are similar to those for texture.



**Note**

The function currently works for ec, ph, and texture. Distribution of locations for texture is similar to those for CEC. The input for this function must be entered in quotation marks. Internet connectivity is needed for the function to work.

**Author(s)**

Christian Thine Omuto

**References**

Batjes, N.H., Ribeiro, E., van Oostrum, A., 2020. Standardised soil profile data to support global mapping and modelling (WoSIS snapshot 2019). *Earth Syst. Sci. Data* 12, 299–320. <https://doi.org/10.5194/essd-12-299-2020>

FAO/IIASA/ISRIC/ISS-CAS/JRC, 2012. Harmonized World Soil Database (version 1.2). FAO and IIASA, Rome.

**See Also**

[DataAvailabilityIndex](#), [PHharm\\_Info](#)

**Examples**

```
SASdata_densityInfo("ec")
```

---

SASglobeData

*A function to query global SAS data*

---

**Description**

A function to query soil data availability in the global SAS database

**Usage**

```
SASglobeData(dframe, ISO, Region)
```

**Arguments**

dframe	is a string to describe type of soil data in the SAS database.
ISO	is a string describing three digit international ISO code for a country.
Region	is a string describing the region of the world.

**Details**

Options for type of soil data for querying the database are "ecse", "ec2", "ec2.5", "ec5", "ph", "phkcl", "phcacl2", "sand", "silt", "c" for regions of the world in the SAS database are "Africa", "Asia", "Europe", "Eurasia", "NENA", "LAC", "N.America", and "Pacific". NENA is Near East and North Africa. LAC is Latin America and Caribbean. N. America is North America. Any of these Regions may be specified if desired. The default Region is NULL

**Value**

The query returns a dataframe with the soil attribute queried, coordinates of sampling locations, and name of country where the samples are located

**Note**

Internet connectivity is needed for the function to work.

**Author(s)**

Christian Thine Omuto

**References**

- Batjes, N. H., Ribeiro, E. & van Oostrum, A. Standardised soil profile data to support global mapping and modelling (WoSIS snapshot 2019). *Earth Syst. Sci. Data* 12, 299–320 (2020).
- FAO/IIASA/ISRIC/ISS-CAS/JRC. Harmonized World Soil Database (version 1.2). (FAO and IIASA, 2012).
- Omuto, C. T., Vargas. R., Abdelmagin, E.A., Mohamed, N., Viatkin, K., Yusuf, Y. Mapping of salt-affected soils – Technical manual. (FAO, 2020). doi:10.4060/ca9215en
- Orgiazzi, A., Ballabio, C., Panagos, P., Jones, A. & Fernández-Ugalde, O. LUCAS Soil, the largest expandable soil dataset for Europe: a review. *Eur J Soil Sci* 69, 140–153 (2018).

**Examples**

```
wrtter=SASglobeData("sand","ZAF")# For South Africa profiles
plot(Longitude~Latitude, wrtter)
```

**Description**

This information function shows the list of EC and pH harmonization models contained in the SAS harmonization service

**Usage**

```
SASmodels(data="ec", extract="1:1")
```

**Arguments**

data	either ec or ph data category for SAS harmonization models
extract	extract solution for measuring ec or ph.

**Details**

This information function shows available models in the SAS harmonization service. The models are divided into two major categories: ec and ph models. The function returns a list of available SAS harmonization models under each data category. ec is the default data category.

**Value**

A list of SAS harmonization models

**Author(s)**

Christian Thine Omuto

**See Also**

[SASglobeData](#), [DataAvailabilityIndex](#)

**Examples**

```
SASmodels("ec", "1:1")
SASmodels("ph", "kcl")
```

---

slopeSuit	<i>A function for assessing slope suitability requirements for certain crops and trees</i>
-----------	--

---

**Description**

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

**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 Beerneart, 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)
splot(slope["carrot"])
```

---

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

---

**Description**

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

**Usage**

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

**Arguments**

<code>ls</code>	length of slope in metres
<code>slope</code>	slope of land in degrees
<code>method</code>	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$LSsmith=sloplenFUN(60,(newmap$slope),"WSmith")
spplot(newmap["LSrenard"])
spplot(newmap["LSsmith"])
```

SOCSuit

*A function for assessing soil carbon suitability requirements for certain crops and trees*

**Description**

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

**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 Beerneart, 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	<i>A function for assessing stoniness suitability requirements for certain crops and trees</i>
-----------	--

---

**Description**

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

**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 Beerneart, 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$fertility,"fertility")
splot(newmap["fertility"])
summary(newmap$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 Beerneart, 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	<i>Sample grid stack map of nutrient indicators for crop fertility requirements</i>
-------------	---

---

### 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 12 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 ... .. .\$ texture1 : Factor w/ 3 levels "Lo","SaLo","SiLo": 2 2 ... ..@ 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	<i>A function to generate georeferenced locations for monitoring soil conditions</i>
--------------	--

---

### 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 the 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(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,texture$texture,"FA0","1:5",soc$soc,clay$clay)
newmap$salt=saltSeverity(newmap$ECe,newmap$ph,0.84,"FA0")
newmap$salineclass=classCode(newmap$salt,"saltseverity")
newmap$salineclass1=as.factor(newmap$salineclass)
splot(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	<i>A function for assessing temperature suitability requirements for certain crops and trees</i>
----------	--

---

### Description

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

### 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 Beerneart, 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 and trees*


---

**Description**

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

**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 Beerneart, 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)
splot(texture["mango"])
```



---

VegCOV	<i>Determining Erosion Cover Factor (C Factor) from Vegetation Cover Fraction</i>
--------	---

---

**Description**

To estimate erosion model cover factor from vegetation cover fraction

**Usage**

```
VegCOV(pcover, model="cai")
```

**Arguments**

pcover	percent vegetation cover
model	model for estimating C factor from percent vegetation cover

**Details**

This function can be used to estimate the C factor for erosion modelling by use of percent vegetation cover. The function contains five models: cai, xu, hurni, jin, and liu. When the percent cover ('pcover') is given and the model to use (e.g., 'cai'), the function returns dimensionless C factor for erosion modelling. C factor ranges between 0 and 1 ( $0 < \text{cfactor} \leq 1$ )

**Value**

A numerical value between 0 and 1

**Note**

The function does not accept percent cover values that are less than 0 and more than 100.

**Author(s)**

Christian Thine Omuto

**References**

C.F. Cai, S.W. Ding, Z.H. Shi, L. Huang, G.Y. Zhang. Study of applying USLE and geographical information system IDRISI to predict soil erosion in small watershed. *J. Soil Water Conserv.*, 14 (2) (2000), pp. 19-24

Taye, G., Vanmaercke, M., Poesen, J., Van Wesemael, B., Tesfaye, S., Teka, D., Nyssen, J., Deckers, J., Haregeweyn, N., 2018. Determining RUSLE P- and C-factors for stone bunds and trenches in rangeland and cropland, North Ethiopia. *Land Degrad Dev* 29, 812–824. <https://doi.org/10.1002/ldr.2814>

**See Also**

[NDVIcoverfactor](#)

**Examples**

```
VegCOV(20,"cai")
```

# Index

## \* datasets

- nutrient, [44](#)
  - nutrindicator, [45](#)
  - soil, [71](#)
  - suitabinput, [76](#)
  - textureinput, [79](#)
- [appendTextureclass, 3, 14](#)
- [carbonateSuit, 4](#)
- [carbonTurnover, 6, 44, 59, 60](#)
- [CECSuit, 7](#)
- [classCode, 8, 10, 56, 61, 62, 64](#)
- [classLUT, 9, 9, 10, 56](#)
- [classnames, 9, 10, 10, 56](#)
- [comparisonTable, 11](#)
- [conversion, 12, 47](#)
- [createTexturedata, 4, 13, 14](#)
- [DataAvailabilityIndex, 14, 65, 67](#)
- [depthharm, 15](#)
- [depthSuit, 16](#)
- [drainageSuit, 18, 48](#)
- [ECconversion1, 12, 19, 21, 22, 24, 39, 40](#)
- [ECconversion2, 12, 20, 20, 22, 24, 39, 40](#)
- [ECconversion3, 20, 21, 22, 57](#)
- [ECconversion4, 12, 20–22, 23, 47](#)
- [ECharm\\_Info, 24, 50](#)
- [ECSuit, 25, 52](#)
- [erodFUN, 9, 26, 28, 29, 48](#)
- [erodibilityRisk, 27, 27](#)
- [erosivFUN, 27, 28, 28](#)
- [ESPSuit, 5, 29, 34, 55](#)
- [featureRep, 31, 36, 52](#)
- [fertilityRating, 32, 34](#)
- [fertilitySuit, 5, 11, 17, 26, 30, 33, 33, 38, 52, 55](#)
- [harmonization, 34](#)
- [imageIndices, 32, 35, 54](#)
- [LGPSuit, 37](#)
- [ME\\_Echarm, 38, 40](#)
- [ME\\_Echarmserve, 35, 39](#)
- [ME\\_PHharm, 39, 40](#)
- [ME\\_PHharmserve, 35, 41](#)
- [NDVIcoverfactor, 41, 81](#)
- [negData, 42](#)
- [NPPmodel, 7, 43, 59, 60](#)
- [nutrient, 44](#)
- [nutrindicator, 45](#)
- [pedoTransfer, 24, 46, 57](#)
- [permeabilityClass, 47](#)
- [PHConversion, 48](#)
- [PHharm\\_Info, 24, 50, 65](#)
- [PHSuit, 19, 26, 38, 51](#)
- [predAccuracy, 52, 54](#)
- [predUncertain, 52, 53, 57](#)
- [rainSuit, 8, 19, 30, 54](#)
- [reclassifyMap, 55](#)
- [regmodelSuit, 54, 56](#)
- [RotCmoistcorrection, 7, 44, 58, 60](#)
- [RotCtempcorrection, 59, 59](#)
- [saltClass, 60, 62, 64](#)
- [saltRating, 33, 61, 61, 64](#)
- [saltSeverity, 61, 62, 63](#)
- [SASdata\\_densityInfo, 24, 50, 64](#)
- [SASglobeData, 65, 67](#)
- [SASmodels, 66](#)
- [slopeSuit, 67](#)
- [sloplenFUN, 27–29, 69](#)
- [SOCSuit, 70](#)
- [soil, 71](#)
- [stoneSuit, 72](#)

suitability, [5](#), [8](#), [11](#), [17](#), [19](#), [26](#), [30](#), [33](#), [34](#),  
[38](#), [52](#), [55](#), [73](#)  
suitabilityClass, [74](#)  
suitabinput, [76](#)  
surveyPoints, [76](#)

tempSuit, [8](#), [17](#), [78](#)  
textureinput, [79](#)  
textureSuit, [4](#), [79](#)

VegCOV, [42](#), [81](#)