Package ‘geotech’

February 14, 2016

Title Geotechnical Engineering

Date 2016-02-01

Version 1.0

Author James Kaklamanos <kaklamanosj@merrimack.edu> and
Kyle Elmy <elmyk@merrimack.edu>

Maintainer James Kaklamanos <kaklamanosj@merrimack.edu>

Description A compilation of functions for performing calculations and creating plots that commonly arise in geotechnical engineering and soil mechanics. The types of calculations that are currently included are: (1) phase diagrams and index parameters, (2) grain-size distributions, (3) plasticity, (4) soil classification, (5) compaction, (6) groundwater, (7) subsurface stresses (geostatic and induced), (8) Mohr circle analyses, (9) consolidation settlement and rate, (10) shear strength, (11) bearing capacity, (12) lateral earth pressures, (13) slope stability, and (14) subsurface explorations. Geotechnical engineering students, educators, researchers, and practitioners will find this package useful.

License GPL-3

Imports graphics, stats

NeedsCompilation no

Repository CRAN

Date/Publication 2016-02-14 17:30:19

R topics documented:

`AASHTO` *AASHTO Soil Classification*

Description

This set of functions classifies soil using the American Association of State Highway and Transportation Officials (AASHTO) Soil Classification System.

Usage

```
AASHTO(sieve = NA, size = NA, percent = NA, metric = NA,
       p10 = NA, p40 = NA, p200 = NA,
       LL = NA, PL = NA, PI = NA, NP = NA)
GI(p200, LL, PL = NA, PI = NA)
```

Arguments

`sieve`	vector of sieve numbers (according to ASTM D422) in grain-size distribution
`size`	vector of grain sizes (in or mm) in distribution
`percent`	vector of percent passing in grain-size distribution
`metric`	logical variable for grain-size distribution: TRUE for metric units (mm), FALSE for English units (in); only required if "size" is supplied
`p10`	percent passing #10 sieve
`p40`	percent passing #40 sieve
`p200`	percent passing #200 sieve
`LL`	liquid limit (percent)
`PL`	plastic limit (percent)
`PI`	plasticity index (percent)
`NP`	logical variable indicating whether the soil is nonplastic (TRUE or FALSE)

Details

In the AASHTO function:

- For the grain-size data, one of three following pieces of data must be specified:
  1. Sieve numbers (sieve); and percent passing
  2. Grain sizes (size); and percent passing
  3. p10, p40, p200
- If sieve data are specified, for sieves larger than the no. 4 sieve, the user should specify the sieve size in inches (e.g., 3/8, 3/4, 1, 1.5, 2, 3, etc.)
- If sieve or size data are specified, this function assumes that the no. 10, 40, and 200 sieves have been used.
- For the plasticity data, one of three following pieces of data must be specified:
  1. LL and PL
  2. LL and PI
  3. NP (if the soil is nonplastic)
- This function only calculates the AASHTO group classification. To obtain the AASHTO group index, use the GI function.

In the GI function, either PL or PI must be specified in addition to the required arguments: F and LL.

Value

- `AASHTO`: AASHTO group classification
- `GI`: AASHTO group index (rounded to the nearest integer)

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

American Association of State and Highway Transportation Officials [AASHTO] (1993). "Recommended Practice for the Classification of Soils and Soil-Aggregate Mixtures for Highway Construction Purposes", AASHTO Designtation 145-91, Standard Specifications for Transportation Materials and Methods of Sampling and Testing.

ASTM (2010). "Annual Book of ASTM Standards", Volume 04.08 – Soil and Rock, ASTM International, Philadelphia.

See Also

`USCS`, `grainSize`, `Plasticity`

Examples

```
## Example code for AASHTO group classification
AASHTO(p10 = 51, p40 = 30, p200 = 15, LL = 40, PI = 10)

## Example code for AASHTO group index
GI(p200 = 48, LL = 45, PI = 21)
```

---

bearingCapacity   *Ultimate Bearing Capacity*

Description

This function computes the ultimate bearing capacity of a shallow foundation using the simple theory of Terzaghi (1943).

Usage

```
bearingCapacity(phi, c, B, L, D, Dw, gamma, gammaW = NA, metric,
                case = "general", shape = "square")
```

Arguments

phi	effective friction angle (deg)
c	effective cohesion (psf or kPa)
B	foundation width (ft or m), or foundation diameter for circular footings
L	foundation length (ft or m)
D	Depth of foundation (ft or m)
Dw	Depth of groundwater table below foundation base (ft or m)
gamma	unit weight of soil (pcf or kN/m^3)
gammaW	unit weight of water (default = 62.4 pcf for English units; 9.81 kN/m^3 for metric units)
case	"general" or "local" to indicate general or local shear failure ("general" is default)
shape	"square", "rectangle", "circle", "strip" (or "continuous")
metric	logical variable: TRUE (for metric units) or FALSE (for English units)

Details

- Either SI or English units can be used, but must stay consistent.
- When specifying the length and width, L should be the longer of the two lengths.
- When the groundwater table is deep or unknown, set Dw >= D.
- For local shear, the friction angle is reduced to a value equal to atan(2/3 * tan(phi)).
- For local shear, the cohesion is reduced to a value equal to 2/3*c.

Value

Bearing capacity (q_ult) from Terzaghi's simple theory (psf or kPa)

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

Terzaghi, K. (1943). Theoretical Soil Mechanics, John Wiley, New York.

See Also

`bearingPressure`, `bearingCapacityFactors`

Examples

```
bearingCapacity(phi = 30, c = 10, B = 10, L = 10, D = 8, Dw = 6,
                gamma = 120, metric = FALSE, case = "local",
                shape = "square")
```

`bearingCapacityFactors` *Bearing Capacity Factors*

Description

Calculate the bearing capacity factors (Nc, Nq, Ngamma) using either the Terzaghi or Vesic methods.

Usage

```
Nq(phi, case = "general", method = "Terzaghi")
Nc(phi, case = "general", method = "Terzaghi")
Ngamma(phi, case = "general", method = "Terzaghi")
```

Arguments

`phi`	friction angle (degrees)
`case`	"general" or "local" to indicate general or local shear failure ("general" is default)
`method`	"Terzaghi" or "Vesic" ("Terzaghi" is default)

Details

- For local shear, the friction angle is reduced to a value equal to atan(2/3 * tan(phi)).
- Ngamma from the Terzaghi method uses the approximate equation of Coduto et al. (2016).

Value

Bearing capacity factor (Nc, Nq, or Ngamma)

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

Coduto, D.P., Kitch, W.A., and Yeung, M.R. (2016). Foundation Design: Principles and Practices, Pearson, Boston.

Terzaghi, K. (1943). Theoretical Soil Mechanics, John Wiley, New York

Vesic, A.S. (1973). Analysis of Ultimate Loads of Shallow Foundations, ASCE Journal of the Soil Mechanics and Foundations Division, Vol. 99, No. SM1, pp. 45-73.

See Also

`bearingCapacity`, `bearingPressure`

Examples

```
Nq(phi = 20, case = "local", method = "Terzaghi")
Nc(phi = 20, case = "local", method = "Terzaghi")
Ngamma(phi = 20, case = "local", method = "Terzaghi")
```

`bearingPressure` *Bearing Pressure*

Description

This function computes the gross bearing pressure that a soil would experience due to a foundation.

Usage

```
bearingPressure(P, B, L, D, Dw, metric, gammaW = NA, gammaC = NA)
```

Arguments

`P`	Vertical gross column load (lb or kN)
`B`	foundation width (ft or m)
`L`	foundation length (ft or m)
`D`	Depth of foundation (ft or m)
`Dw`	Depth of groundwater table below foundation base (ft or m)
`metric`	logical variable: TRUE (for metric units) or FALSE (for English units)
`gammaW`	unit weight of water (default = 62.4 pcf for English units; 9.81 kN/m^3 for metric units)
`gammaC`	unit weight of concrete (default = 150 pcf for English units; 23.6 kN/m^3 for metric units)

Details

- Either SI or English units can be used, but must stay consistent.
- When specifying the length and width, L should be the longer of the two lengths.
- For a continuous (strip) foundation, specify L = 1 and specify P as the load per unit length.
- When the groundwater table is deep or unknown, set Dw >= D.

Value

Gross bearing pressure of footing (psf or kPa)

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

See Also

bearingPressure, bearingCapacityFactors

Examples

```
##  Calculation of bearing pressure under a rectangular footing
bearingPressure(P = 1000, B = 2, L = 5, D = 6, Dw = 2, metric = FALSE)
```

---

grainSize — *Grain-Size Distribution Functions*

Description

These functions compute diferent aspects related to grain-size distributions of soil.

Usage

```
size.from.sieve(sieve, metric)
grainSize.plot(sieve = NA, size = NA, percent, metric)
percentComponents(sieve = NA, size = NA, percent, metric)
Dsize(N, sieve = NA, size = NA, percent, metric)
grainSize.coefs(percent, sieve = NA, size = NA, D10 = NA,
                D30 = NA, D60 = NA)
```

Arguments

sieve	vector of sieve numbers according to ASTM D422
size	vector of grain sizes (in or mm)
percent	vector of percent passing
metric	logical variable: TRUE for metric units (mm), FALSE for English units (in)
N	the percent corresponding to the desired D-size

`D10`	D-size corresponding to 10 percent passing
`D30`	D-size corresponding to 30 percent passing
`D60`	D-size corresponding to 60 percent passing

Details

- Either sieve numbers (`sieve`) OR grain sizes (`size`) must be provided in all functions; however, in grainSize.coefs the user additionally has the option of specifying D10, D30, and D60.
- For sieves larger than the no. 4 sieve, the user should specify the sieve size in inches (e.g., 3/8, 3/4, 1, 1.5, 2, 3, etc.).
- The argument `percent` is required in all functions except size.from.sieve, and the argument `metric` is required in all functions except for grainSize.coefs and Dsize (although it is needed in Dsize if sieve numbers are provided).
- The percentComponents function assumes that the no. 4 and no. 200 sieves have been used.

Value

- sieve.from.size calculates a set of grain sizes corresponding to a set of sieves (output is vector of grain sizes (in or mm)
- grainSize.plot creates a plot of the soil's grain-size distribution; no numerical values are output
- percentComponents is used to calculate the percent gravel, sand, and fines, and outputs a three-element list:
  1. pg = Percent gravel
  2. ps = percent sand
  3. pf = percent fines
- Dsize outputs the grain size corresponding to a certain percent finer (N), given a grain-size distribution
- grainSize.coefs calculates the coefficients of uniformity and curvature, and outputs a two-element list:
  1. Cu = Coefficient of uniformity (D60 / D10)
  2. Cc = Coefficient of Curavture (D30^2 / (D10 * D60))

Note

The Dsize function uses logarithmic interpolation to calculate the D-size from the provided grain-size distribution. Log-linear extrapolation is used for grain sizes beyond the range of the data, and a warning is provided.

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

See Also

`USCS`, `AASHTO`, `Plasticity`

Examples

```
##  Example 1:  Grain-size distribution

##  (a) Define data
sieve.example <- c(3/8, 4, 10, 20, 40, 140, 200)
percent.example <- c(95.72, 90.23, 81.49, 66.36, 50.00, 8.51, 4.82)

##  (b) Percent gravel, sand, and fines
percentComponents(sieve = sieve.example, percent = percent.example,
                  metric = TRUE)

##  (c) Plot grain-size distribution
grainSize.plot(sieve = sieve.example, percent = percent.example,
               metric = TRUE)

##  (d) Calculate D50
Dsize(N = 50, sieve = sieve.example, percent = percent.example,
      metric = TRUE)

##  (e) Coefficients of uniformity and curvature
grainSize.coefs(sieve = sieve.example, percent = percent.example)


##  Example 2:  coefficients of uniformity and curvature
grainSize.coefs(D60 = 0.10, D30 = 0.03, D10 = 0.002)
```

`hydraulicConductivity` *Hydraulic Conductivity*

Description

These functions are used to calculate a soil's hydraulic conductivity from standard laboratory tests (`kConstant` for the constant head test and `kFalling` for the falling head test), and to calculate the equivalent horizontal and vertical hydraulic conductivity for layered soil deposits (`kx` and `kz`, respectively).

Usage

```
kConstant(V, t, h, L, As = NA, Ds = NA)
kFalling(h0, hf, t, L, As = NA, Ap = NA, Ds = NA, Dp = NA)
kx(thk, k)
kz(thk, k)
```

Arguments

`t`	time of flow
`L`	length of soil sample

`As`	cross-sectional area of the soil sample
`Ds`	diameter of soil sample
`V`	volume of water collected (constant head test)
`h`	head difference between inflow and outflow (constant head test)
`h0`	head difference at beginning of test (falling head test)
`hf`	head difference at end of test (h0 > hf; falling head test)
`Ap`	cross-sectional area of the standpipe (falling head test)
`Dp`	diameter of the standpipe (falling head test)
`thk`	vector of layer thicknesses
`k`	vector of hydraulic conductivities

Details

- Either English or metric units can be used, but they must be consistent.
- Either the areas or the diameters of the soil sample (or standpipe) need to be specified.

Value

- kConstant calculates the measured hydraulic conductivity from the constant head test
- kFalling calculates the measured hydraulic conductivity from the falling head test
- kx calculates the equivalent hydraulic conductivity in the horizontal direction for a layered soil deposit
- kz calculates the equivalent hydraulic conductivity in the vertical direction for a layered soil deposit

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

See Also

`wellHydraulics`

Examples

```
##  Example code for Hydraulic Conductivity from Constant head test
##  k in units of cm/s
kConstant(V = 800, t = 100, h = 200, A = 40, L = 50)

##  Example code for Hydraulic Conductivity from Falling head test
##  k in units of cm/s
kFalling(h0 = 12, hf = 2, L = 10, Ds = 20, Dp = 2, t = 100)
```

indexParameters *Phase Diagrams and Index Parameters*

Description

These functions compute and plot phase diagrams and index parameters associated with a soil's composition.

Usage

```
phase.plot(Ws, Ww, Vs, Vw, Va, W.unit, V.unit, mass = FALSE)
phase.params(Ws, Ww, Vs, Vw, Va)
waterContent(M1, M2, Mc)
relDensity(e, emax, emin)
```

Arguments

Ws	Weight of solids
Ww	Weight of water
Vs	Volume of solids
Vw	Volume of water
Va	Volume of air
W.unit	Measurement unit of weights
V.unit	Measurement unit of volume
mass	logical variable: TRUE for masses or FALSE for weights (default)
M1	Mass (or weight) of can and wet soil, before drying in oven
M2	Mass (or weight) of can and dry soil, after drying in oven
Mc	Mass (or weight) of can
e	Void ratio
emax	Maximum void ratio
emin	Minimum void ratio

Details

- In phase.plot, if any parameters are zero, please enter "0"; do not leave them blank.
- In waterContent, either masses or weights may be used, because the units cancel.

Value

- `phase.plot` plots phase diagrams from weights (or masses) and volumes of a soil sample. No numerical output is provided.
- `phase.params` calculates a ten-element list of index parameters from provided weights and volumes of a soil sample:
  1. w = water content (as decimal)
  2. S = degree of saturation (as decimal)
  3. e = void ratio (as decimal)
  4. n = porosity (as decimal)
  5. Gs = specific gravity
  6. gammaT = total unit weight
  7. gammaD = dry unit weight
  8. gammaS = unit weight of solids
  9. gammaW = unit weight of water
  10. gammaB = buoyant unit weight
- `waterContent` calculates the water content (as a decimal) from lab results (i.e., measured weights or masses)
- `relDensity` calculates the relative density (as a decimal) from specified void ratios (e, emax, and emin).

Note

The phase diagram in phase.plot is currently not to scale; this may be edited in the future to allow this functionality.

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

Examples

```
##  Example of phase diagram plot:
phase.plot(Ws = 75.8, Ww = 15.6, Vs = 0.45, Vw = 0.25,
           Va = 0.1, W.unit = "lb", V.unit = "ft^3", mass = FALSE)

##  Example of index parameters function:
phase.params(Ws = 75.8, Ww = 15.6, Vs = 0.45, Vw = 0.25, Va = 0.1)

##  Example of water content function:
waterContent(M1 = 20.68, M2 = 18.14, Mc = 8.20)

##  Example of relative density function:
relDensity(e = 0.3, emax = 0.92, emin = 0.35)
```

`lateralEarthPressures` *Lateral Earth Pressures*

Description

These functions are used to calculate lateral earth pressure coefficients using different methods.

Usage

```
K(sigmax, sigmaz)
Ko(phi, OCR = 1)
Ka(phi, beta = 0)
Kp(phi, beta = 0)
```

Arguments

`sigmax`	horizontal effective stress
`sigmaz`	vertical effective stress
`phi`	effective friction angle (deg)
`OCR`	overconsolidation ratio; default = 1
`beta`	angle of backfill (deg); default = 0

Details

These functions either compute the lateral earth pressure coefficient from provided stresses (K), common empirical correlations (Ka), or basic lateral earth pressure theories (Kp).

- For Ko, the Jaky (1944) equation is used for normally consolidated soil, and the Mayne and Kulhawy (1982) equation is used for overconsolidated soil.
- For Ka and Kp, the basic Rankine (1857) theory is used. Note that this theory requires beta <= phi.

Value

- K = Coefficient of lateral earth pressure (directly from provided stresses)
- Ko = Coefficient of lateral earth pressure at rest
- Ka = Coefficient of lateral earth pressure (active)
- Kp = Coefficient of lateral earth pressure (passive)

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

Jaky, J. (1944). "The Coefficient of Earth Pressure at Rest", Journal for the Society of Hungarian Architects and Engineers, October, pp. 355-358.

Mayne, P.W., and Kulhawy, F.H. (1982). "Ko-OCR Relationships in Soil", Journal of the Geotechnical Engineering Division, ASCE, Vol. 108, No. GT6, pp. 851-872.

Rankine, W.J.M. (1857). "On the Stability of Loose Earth", Philosophical Transactions of the Royal Society, Vol. 147, London.

See Also

`stressHorizontal`

Examples

```
##  Example code for Coefficient of lateral earth pressure
K(sigmax = 50, sigmaz = 90)

##  Example code for Coefficient of lateral earth pressure at rest
Ko(phi = 20, OCR = 1)

##  Example code for Coefficient of lateral earth pressure (active)
Ka(phi = 30, beta = 10)

##  Example code for Coefficient of lateral earth pressure (passive)
Kp(phi = 30, beta = 10)
```

`MohrCircle` *Mohr Circle Calculations*

Description

These functions are used to calculate the different parameters associated with Mohr Circle:

- `sigmaTrans` performs stress-transformation calculations at a specific angle of inclination
- `MohrCircle.calc` performs calculations associated with the Mohr Circle (stress transformation over a range of angles)
- `MohrCircle.plot` creates a plot of the Mohr Circle for a given state of stress
- `sigma13` calculates the magnitudes and orientations of the principal stresses
- `tauMax` calculates the magnitude and orientation of the maximum in-plane shear stress

Usage

```
sigmaTrans(theta, sigmaX = NA, sigmaZ = NA, tauXZ = NA,
           sigma1 = NA, sigma3 = NA)
MohrCircle.calc(sigmaX = NA, sigmaZ = NA, tauXZ = NA,
                sigma1 = NA, sigma3 = NA,
                theta = seq(from = 0, to = 180, by = 1))
MohrCircle.plot(sigmaX = NA, sigmaZ = NA, tauXZ = NA,
                sigma1 = NA, sigma3 = NA, metric = TRUE)
sigma13(sigmaX, sigmaZ, tauXZ)
tauMax(sigmaX, sigmaZ, tauXZ)
```

Arguments

`sigmaX`	normal stress acting in the horizontal direction
`sigmaZ`	normal stress acting in the vertical direction
`tauXZ`	shear stress acting on the same plane as sigmaX
`sigma1`	major principal stress
`sigma3`	minor principal stress
`theta`	angle of inclination (degrees) [see details below]
`metric`	units, if TRUE units are SI, if FALSE units are English

Details

- For sigmaTrans, MohrCircle.calc, and MohrCircle.plot, One of the following two sets of data must be entered:
  1. sigmaX, sigmaZ, and tauXZ
  2. sigma1 and sigma3
- In functions sigmaTrans and MohrCircle.calc, if theta is entered in conjunction with sigmaX, sigmaZ, and tauXZ, it is interpreted as the angle of inclination above the horizontal. If theta is entered in conjunction with the principal stresses, then it is interpreted as the angle of inclination above the major principal plane.
- Note that theta is required for sigmaTrans, optional for MohrCircle.calc, and not used in MohrCircle.plot, sigma13, and tauMax.

Value

- `sigmaTrans` outputs a two-element list containing the results of the stress-transformation calculations:
  1. sigma = normal stress on an inclined plane
  2. tau = shear stress on an inclined plane
- `MohrCircle.calc` outputs a five-element list containing Mohr Circle calculations:
  1. C = center of Mohr circle
  2. R = radius of Mohr circle
  3. sigma = vector of normal stresses for Mohr circle

4. tau = vector of shear stresses for Mohr circle
5. theta = vector of angles (deg)

- `MohrCircle.plot` produces a plot of Mohr Circle; no numerical calculations are output from the function.
- `sigma13` outputs a four-element list containing the magnitudes and directions of the principal stresses:
  1. sigma1 = magnitude of major principal stress
  2. sigma3 = magnitude of minor principal stress
  3. theta1 = direction of major principal stress (deg)
  4. theta3 = direction of minor principal stress (deg)
- `tauMax` outputs a two-element list containing the magnitude and direction of the maximum in-plane shear stress:
  1. tauMax = maximum in-plane shear stress
  2. theta = angle of maximum in-plane shear stress

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

Examples

```
##  Example code for Stress Transformation
sigmaTrans(sigmaX = 80, sigmaZ = 120, tauXZ = 20, theta = 78)

##  Example code for Mohr Circle Calculations
MohrCircle.calc(theta = 20, sigmaX = 80, sigmaZ = 120, tauXZ = 20)

##  Example code for Mohr Circle Plot
MohrCircle.plot(sigmaX = 80, sigmaZ = 120, tauXZ = 20, metric = FALSE)

##  Example code for Principal Stresses
sigma13(sigmaX = 80, sigmaZ = 120, tauXZ = 20)

##  Example code for Maximum In-Plane Shear Stress
tauMax(sigmaX = 80, sigmaZ = 120, tauXZ = 20)
```

`Plasticity` *Plasticity Functions*

Description

These functions compute diferent aspects related to plasticity:

- `LL` calculates and plots a soil's liquid limit (LL) using the flow curve from measured lab data.
- `PI` calculates plasticity index.

- `LI` calculates liquidity index.
- `A.line` calculates the PI corresponding to the A-line on Casagrande's plasticity chart.
- `plasticity.plot` plots a soil's plasticity parameters on Casagrande's plasticity chart.

Usage

```
LL(N, w, draw)
PI(LL, PL)
LI(w, LL, PL)
A.line(LL)
plasticity.plot(LL, PL = NA, PI = NA)
```

Arguments

`LL`	Liquid Limit (percent)
`PL`	Plastic Limit (percent)
`PI`	Plasticity index (percent)
`w`	moisture content (percent) [see details below]
`N`	vector of number of blows obtained from the liquid limit test (for LL function)
`draw`	logical variable: TRUE if plot of flow curve is desired, and FALSE to suppress the creation of the plot (for LL function)

Details

- Note that all inputs should be entered as percents, not decimals (e.g. w = 23, not w = 0.23)
- The argument w (water content) should be a single value for the LI function that represents the soil's in-situ moisture content; and w should be a vector of values for the LL function that represent the measured water contents, each with N blows.
- For plot.plasticity and A.line, either PL or PI must be specified in addition to LL.

Value

- `LL` outputs a soil's liquid limit and, optionally, a plot of the soil's flow curve.
- `PI` outputs a two-element list composed of:
  1. PI = Plasticity index (percent)
  2. descr = qualitative description of soil based on Sowers (1979)
- `LI` outputs a two-element list composed of:
  1. LI = Liquidity index (percent)
  2. descr = qualitative description of soil based on Sowers (1979)
- `A.line` outputs the plasticity index corresponding to the A-line on Casagrande's plasticity chart
- `plasticity.plot` is used to make a plot of a soil's plasticity parameters (LL and PI) on Casagrande's plasticity chart. No numerical values are output from this function.

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

Casagrande, A.M. (1948). "Classification and Identification of Soils", ASCE Transactions, Vol. 113, pp. 901-991.

Sowers, G.F. (1979). "Introductory Soil Mechanics and Foundations: Geotechnical Engineering", 4th edition, Macmillan, New York.

See Also

`USCS`, `AASHTO`, `grainSize`

Examples

```
## Example code for Plasticity Index
PI(LL = 80, PL = 30)

## Example code for Liquidity Index
LI(w = 55, PL = 20, LL = 50)

## Example code for plasticity plot
plasticity.plot(LL = 40, PL = 20)

## Example code for liquid limit analysis
LL(N = c(72, 37, 14), w = c(7, 15, 21), draw = TRUE)
```

`slopeStability` *Slope Stability*

Description

These functions are used to determine the factor of safety against shear failure on slopes using infinite slope analyses (FSinf) or planar failure analyses (FSplanar).

Usage

```
FSinf(c, phi, gamma, gammaW = NA, alpha, D, zw, metric)
FSplanar(c, phi, alpha, L, W, u)
```

Arguments

`c`	soil cohesion
`phi`	soil friction angle (degrees)
`gamma`	soil unit weight
`gammaW`	unit weight of water (default = 62.4 pcf for English units; 9.81 kN/m^3 for metric units)
`alpha`	slope angle (angle of failure plane) for infinite slope analysis; angle of failure plane for planar failure analysis (deg)
`D`	depth to failure plane
`zw`	distance of groundwater table above failure plane (use 0 for a dry slope and D for a submerged slope with parallel seepage)
`metric`	logical variable: TRUE (for metric units: kN/m^3) or FALSE (for English units: pcf) [this is needed if gammaW is unspecified]
`L`	length of failure plane (planar failure analysis)
`W`	weight of slope per unit width (planar failure analysis)
`u`	average pressure head on the failure plane (planar failure analysis)

Details

- The assumptions of infinite slope analyses include (Coduto et al., 2011):
  1. The slope face is planar and of infinite extent.
  2. The failure surface is parallel to the slope face.
  3. Vertical columns of equal dimensions through the slope are identical.
- For planar failure analysis, the angle of the failure plane is generally not equal to the failure angle. The geometry of the failure wedge is specified using its weight, W.
- Either English or metric units can be used, but must be consistent.

Note

See a geotechnical engineering textbook such as Coduto et al. (2011) for more background on slope stability theory.

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

Coduto, D.P., Yeung, M.R., and Kitch, W.A. (2011). Geotechnical Engineering: Principles and Practices, Second Edition. Pearson: Upper Saddle River, N.J.

`soilProfile` *Plot of Soil Profile*

Description

This function creates a plot of a soil profile.

Usage

```
soil.profile(thk = NA, depth = NA, zw, type = NA, gamma = NA,
                     phi = NA, C = NA, title = "Soil Profile", metric)
```

Arguments

`thk`	vector of layer thicknesses (ft or m)
`depth`	vector of layer bottom depths (ft or m)
`zw`	depth of groundwater table (ft or m)
`type`	vector of soil types (character strings)
`gamma`	vector of unit weights (pcf or kN/m^3)
`phi`	vector of soil friction angles (deg)
`C`	vector of soil cohesion (psf or kPa)
`title`	desired title of plot (deafult: "Soil Profile")
`metric`	logical variable: TRUE (for metric units) or FALSE (for English units)

Details

- Either layer thicknesses or depth to layer bottoms must be specified.
- The only necessary variables are thk (or depth), zw, and metric. All other variables are optional.

Value

This function creates a plot of a soil profile; no numerical values are returned.

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

Examples

```
##  Example code for plot of a soil profile
soil.profile(depth = c(20, 40, 52, 60), zw = 20, type = c("Dry Sand",
            "Saturated Sand", "Soft Clay", "Dense Gravel"),
            gamma = c(110, 115, 120, 150), phi = c(30, 30, NA, 38),
            C = c(NA, NA, 300, NA), metric = FALSE)
```

SPT — *Standard Penetration Test (SPT) Corrected N-values*

Description

These functions are used to calculate corrected blow counts (N-values) for the Standard Penetration Test (SPT). Function N60 calculates corrections for field procedures, and function N160 calculates corrections for field procedures and overburden pressure.

Usage

```
N60(N, Lr, Db, SS = TRUE, E = 0.60, metric)
N160(N60, sigma, metric)
```

Arguments

`N`	raw SPT N-value
`Lr`	rod length (ft or m)
`E`	hammer efficiency as a decimal (default: 0.60)
`Db`	borehole diameter (in or mm)
`SS`	logical variable: TRUE for standard sampler [default]; FALSE for sampler without liner
`N60`	SPT N-value corrected for field procedures
`sigma`	effective vertical stress at the depth of interest (psf or kPa)
`metric`	logical variable: TRUE (for metric units) or FALSE (for English units)

Value

- N60 = SPT blow count (N-value) corrected for field procedures, after Skempton (1986)
- N1,60 = SPT blow count (N-value) corrected for field procedures and overburden pressure, using the Liao and Whitman (1986) stress correction

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

Liao, S.S.C., and Whitman, R.V. (1986). "Overburden Correction Factors for SPT in Sand", Journal of Geotechnical Engineering, Vol. 112, No. 3, pp. 373-377, ASCE.

Skempton, A.W. (1986). "Standard Penetration Test Procedures and the Effects in Sands of Overburden Pressure, Relative Density, Particle Size, Aging, and Overconsolidation", Geotechnique, Vol. 36, No. 3, pp. 425-447.

Examples

```
##  Example code for SPT blow count (N-value) corrected for field
##  procedures
N60(N = 11, Lr = 25, Db = 4, E = 0.50, SS = TRUE, metric = FALSE)

##  Example code for SPT blow count (N-value) corrected for field
##  procedures and overburden pressure
##  In English units
N160(N60 = 8, sigma = 1500, metric = FALSE)
##  In SI units
N160(N60 = 8, sigma = 90, metric = TRUE)
```

---

stressHorizontal *Horizontal Stress Calculations*

Description

These functions calculate horizontal stress at a point (sigmaX) and versus depth (sigmaX.profile)

Usage

```
sigmaX(gamma, thk = NA, depth = NA, zw, zout, K, gammaW = NA,
       metric, upper = TRUE)
sigmaX.profile(gamma, thk = NA, depth = NA, K, zw, zout = NA,
               gammaW = NA, metric)
```

Arguments

gamma	vector of unit weights (pcf or kN/m^3)
thk	vector of layer thicknesses (ft or m)
depth	vector of layer bottom depths (ft or m)
zw	depth of groundwater table (ft or m)
zout	desired depth of output (ft or m)
K	vector of lateral earth pressure coefficients
gammaW	unit weight of water (default = 62.4 pcf for English units;9.81 kN/m^3 for metric units)
metric	logical variable: TRUE (for metric units) or FALSE (for English units)
upper	logical variable when using the sigmaX function to specify whether the upper (TRUE) or lower (FALSE) lateral earth pressure coefficient should be used, for the special case that zout corresponds to a layer interface

Details

- Either layer thicknesses or depths to layer bottoms must be specified.
- The argument zout should be a single value for sigmaX, and a vector of values for sigmaX.profile. For sigmaX.profile, zout defaults to critical locations in the profile (the top and bottom of the profile, layer interfaces, and the groundwater table) [recommended].

Value

Function sigmaX outputs a three-element list giving the stresses at a specified depth:

- sigmaX.eff = effective horizontal stress
- sigmaX.total = total horizontal stress
- u = pore water pressure

Function sigmaX.profile outputs a four-element list giving the stress variation with depth (four vectors):

- depth = depth
- sigmaX.eff = effective horizontal stress
- sigmaX.total = total horizontal stress
- u = pore water pressure

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

See Also

lateralEarthPressures, stressPlot, stressVertical

Examples

```
##  Example code for Horizontal Stress at a point
sigmaX(gamma = c(108, 116), depth = c(15, 40), zout = 18,
       K = c(0.34, 0.32), zw = 15, metric = FALSE, upper = TRUE)

##  Example code for Horizontal Stress Profile
sigmaX.profile(gamma = c(108, 116), depth = c(15, 40),
               K = c(0.34, 0.32), zw = 15, metric = FALSE)
```

`stressInducedArea` *Induced Stress due to Area Loads*

Description

These functions calculate induced stresses due to an area load applied at the surface: induced.area performs this calculation at a specific depth, induced.area.profile performs this calculation at a series of depths.

Usage

```
induced.area(z, q, B, L = NA, shape)
induced.area.profile(z = NA, q, B, L = NA, shape)
```

Arguments

`z`	depth(s) of interest
`q`	applied pressure at ground surface
`B`	width of loaded area
`L`	length of loaded area (rectangular foundations only)
`shape`	shape of loaded area (a string containing "circle", "square", "strip", or "rectangle")

Details

- The depth(s) of interest (z) should be a single value for induced.point, and a vector of values for induced.point.profile. For induced.point.profile, the default vector of z values is 1-ft or 1-m increments, to a maximum depth of 50 ft or 50 m.
- This function currently uses the approximate method of Poulos and Davis (1974). More advanced formulations are expected in future versions of this package.

Value

Function induced.area outputs the induced vertical stress at the center of the loaded area.

Function induced.area.profile outputs a two-element list containing two vectors:

- depth = vector of depths
- sigmaZ = vector of induced vertical stresses below center of the loaded area

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

Poulos, H.G., and Davis, E.H. (1974). Elastic Solutions for Soil and Rock Mechanics, John Wiley, New York.

See Also

stressInducedPoint, stressVertical, stressHorizontal

Examples

```
##  Example code for Induced Stress due to Area Load
induced.area(z = 10, q = 1000, B = 3, shape = "square")

##  Example code for Induced Stress due to Area Load: Profile
induced.area.profile(q = 1000, B = 3, shape = "square")
```

---

stressInducedPoint   *Induced Stress due to Point Loads*

Description

These functions calculate induced stresses due to a point load applied at the surface: induced.point performs this calculation at a specific depth, induced.point.profile performs this calculation at a series of depths.

Usage

```
induced.point(P, x, y, z, nu)
induced.point.profile(P, x, y, z = NA, nu)
```

Arguments

P	point load
x	horizontal distance from point load in the x direction
y	horizontal distance from point load in the y direction
z	depth(s) of interest
nu	Poisson's ratio

Details

The depth(s) of interest (z) should be a single value for induced.point, and a vector of values for induced.point.profile. For induced.point.profile, the default vector of z values is 1-ft or 1-m increments, to a maximum depth of 50 ft or 50 m.

Value

Function induced.point outputs a six-element list giving the induced stress at a specified depth using the Boussinesq (1885) theory.

- sigmaX = induced horizontal stress in the x direction
- sigmaY = induced horizontal stress in the y direction

- sigmaZ = induced vertical stress
- tauZX = induced shear stress on the ZX plane
- tauYX = induced shear stress on the YX plane
- tauYZ = induced shear stress on the YZ plane

Function induced.point.profile outputs a seven-element list containing seven vectors that display the depth variation of induced stresses using the Boussinesq (1885) theory.

- depth = depth
- sigmaX = induced horizontal stress in the x direction
- sigmaY = induced horizontal stress in the y direction
- sigmaZ = induced vertical stress
- tauZX = induced shear stress on the ZX plane
- tauYX = induced shear stress on the YX plane
- tauYZ = induced shear stress on the YZ plane

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

Boussinesq, J. (1885). "Application des Potentiels a L'Etude de L'Equilibre et du Mouvement des Solides Elastiques", Gauthier-Villars, Paris (in French).

See Also

stressInducedArea, stressVertical, stressHorizontal

Examples

```
##  Example code for Induced Stress due to a Point Load
induced.point(P = 100000, x = 5, y = 2, z = 6, nu = 0.35)

##  Example code for Induced Stress due to a Point Load: Profile
induced.point.profile(P = 100000, x = 5, y = 2, nu = 0.35)
```

stressPlot *Plot of Stress Profile*

Description

These functions produce plots of vertical stresses versus depth (plot.sigmaZ) or horizontal stresses versus depth (plot.sigmaX).

Usage

```
sigmaX.plot(depth, sigmaX.eff, sigmaX.total = NA, u = NA, metric)
sigmaZ.plot(depth, sigmaZ.eff, sigmaZ.total = NA, u = NA, metric)
```

Arguments

`depth`	vector of depths (ft or m)
`sigmaX.eff`	vector of effective horizontal stresses (psf or kPa)
`sigmaX.total`	vector of total horizontal stresses (psf or kPa)
`sigmaZ.eff`	vector of effective vertical stresses (psf or kPa)
`sigmaZ.total`	vector of total vertical stresses (psf or kPa)
`u`	vector of effective vertical stresses (psf or kPa)
`metric`	logical variable: TRUE (for metric units) or FALSE (for English units)vector of total vertical stresses (psf or kPa)

Details

- Arguments sigmaX.eff and sigmaX.total are used in conjunction with sigmaX.plot, and arguments sigmaZ.eff and sigmaZ.total are used in conjunction with sigmaZ.plot.
- If total stresses and pore water pressure are left blank, the plot is only constructed for effective stress.
- Once constructed, additional profiles may be added by the user to this plot (for example, for induced stress or maximum past pressure).

Value

This function creates a plot of stresses; no numerical output is obtained. To perform numerical stress calculations, see `stressHorizontal` or `stressVertical`.

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

See Also

`stressHorizontal`, `stressVertical`

Examples

```
##  Example code for vertical stress plot
temp <- sigmaZ.profile(gamma = rep(100, 3), depth = c(10, 20, 30),
                       zw = 10, metric = FALSE)
depth <- temp$depth
sigmaTotal <- temp$sigmaZ.total
u <- temp$u
sigmaEff <- temp$sigmaZ.eff
sigmaZ.plot(depth = depth, sigmaZ.eff = sigmaEff, metric = FALSE,
            sigmaZ.total = sigmaTotal, u = u)

##  Example code for horizontal stress plot
##  Site with constant unit weight = 100 pcf, GWT at 10 ft depth
temp <- sigmaX.profile(gamma = rep(100, 3), depth = c(10, 20, 30),
                       K = c(0.35, 0.30, 0.28), zw = 10, metric = FALSE)
depth <- temp$depth
sigmaTotal <- temp$sigmaX.total
u <- temp$u
sigmaEff <- temp$sigmaX.eff
sigmaX.plot(depth = depth, sigmaX.eff = sigmaEff, metric = FALSE,
            sigmaX.total = sigmaTotal, u = u)
```

---

stressVertical *Vertical Stress Calculations*

Description

These functions calculate vertical stress at a point (sigmaZ) and versus depth (sigmaZ.profile)

Usage

```
sigmaZ(gamma, thk = NA, depth = NA, zw, zout, gammaW = NA, metric)
sigmaZ.profile(gamma, thk = NA, depth = NA, zw, zout = NA,
                gammaW = NA, metric)
```

Arguments

`gamma`	vector of unit weights (pcf or kN/m^3)
`thk`	vector of layer thicknesses (ft or m)
`depth`	vector of layer bottom depths (ft or m)
`zw`	depth of groundwater table (ft or m)
`zout`	desired depth of output (ft or m): a single value
`gammaW`	unit weight of water (default = 62.4 pcf for English units; 9.81 kN/m^3 for metric units)
`metric`	logical variable: TRUE (for metric units) or FALSE (for English units)

Details

Either layer thicknesses or depths to layer bottoms must be specified.

Value

Function sigmaZ outputs a three-element list giving the stresses at a specified depth:

- sigmaZ.eff = effective vertical stress
- sigmaZ.total = total vertical stress
- u = pore water pressure

Function sigmaZ.profile outputs a four-element list giving the stress variation with depth (four vectors):

- depth = depth
- sigmaZ.eff = effective vertical stress
- sigmaZ.total = total vertical stress
- u = pore water pressure

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

See Also

stressHorizontal, stressInducedArea, stressInducedPoint, stressPlot

Examples

```
## Example code for Vertical Stress at a Point
sigmaZ(gamma = c(108, 116), depth = c(15, 40), zout = 18, zw = 15, metric = FALSE)

##  Example code for Vertical Stress Profile
sigmaZ.profile(gamma = c(108, 116), depth = c(15, 40), zw = 15, metric = FALSE)
```

USCS — *USCS Soil Classification*

Description

This set of functions classifies soil using the Unified Soil Classification System (USCS).

Usage

```
USCS(pg = NA, ps = NA, pf = NA, Cc = NA, Cu = NA,
     LL = NA, PL = NA, PI = NA, sieve = NA, size = NA,
     percent = NA, metric = NA)
USCS.fine.symbol(LL, PL, PI = NA)
USCS.coarse.symbol(pg, ps, pf, Cc, Cu, PI = NA, LL, PL)
```

Arguments

`pg`	percent gravel
`ps`	percent sand
`pf`	percent fines
`Cu`	coefficient of uniformity
`Cc`	coefficient of curvature
`sieve`	vector of sieve numbers (according to ASTM D422) in grain-size distribution
`size`	vector of grain sizes (in or mm) in distribution
`percent`	vector of percent passing in grain-size distribution
`metric`	logical variable for grain-size distribution: TRUE for metric units (mm), FALSE for English units (in); only required if "size" is supplied
`LL`	liquid limit (percent)
`PL`	plastic limit (percent)
`PI`	plasticity index (percent)

Details

The USCS function is the master function for performing soil classifications:

- Data on the soil's grain-size distribution are required if the percent fines is less than or equal to 12 percent. The user has three options for input to this function:
  1. Sieve numbers (sieve); and percent passing
  2. Grain sizes (size); and percent passing
  3. Percent components (pg, ps, pf) and coefficients of uniformity and curvature (Cc and Cu)
- Data on the soil's fines [either (a) LL and PL or (b) LL and PI] are required if the percent fines is greater than or equal to 5 percent.
- If percent components are specified, then Cu and Cc are needed if pf < 85 percent.

The USCS.fine.symbol and USCS.coarse.symbol are used to obtain the group symbols for fine-grained and coarse-grained soils, respectively. For USCS.fine.symbol, either PI or PL must be specified in addition to LL. For USCS.coarse.symbol:

- Data on the soil's gradation (Cu and Cc) are required if the percent fines is less than or equal to 12 percent.
- Data on the soil's fines [either (a) LL and PL or (b) LL and PI] are required if the percent fines is greater than or equal to 5 percent.

Value

- USCS outputs a two-element list providing a soil's USCS group symbol and name:
  1. symbol = USCS group symbol
  2. name = USCS group name
- USCS.fine.symbol outputs a fine-grained soil's two-letter group symbol.
- USCS.coarse.sybmol outputs a coarse-grained soil's two-letter group symbol.

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

References

ASTM (2010). "Annual Book of ASTM Standards", Volume 04.08 – Soil and Rock, ASTM International, Philadelphia.

See Also

AASHTO, grainSize, Plasticity

Examples

```
## Example code for USCS
USCS(pg = 15, ps = 34, pf = 51, Cc = 1, Cu = 4, LL = 40, PL = 10)
```

---

wellHydraulics *Well Hydraulics*

Description

These functions are used to calculate flow rate to wells (`wellFlow`), drawdown to wells (`wellDrawdown`), and hydraulic conductivity from pumping tests (`kPump`).

Usage

```
wellFlow(k, H = NA, h0, hf, r0, rw)
wellDrawdown(Q, k, H = NA, h0, r0, rw, r)
kPump(Q, H = NA, h1, h2, r1, r2)
```

Arguments

`Q`	flow rate into well
`k`	hydraulic conductivity of aquifer
`H`	thickness of aquifer
`h0`	initial total head in aquifer (before pumping)
`hf`	final total head in well casing (after pumping)
`r0`	radius of influence
`rw`	radius of well
`r`	radius of interest (for drawdown calculations)
`h1`	total head in farthest observation well (kPump
`h2`	total head in nearest observation well (kPump)
`r1`	radius from pumped well to farthest observation well (kPump)
`r2`	radius from pumped well to nearest observation well (kPump)

Details

- Datum for total heads is the bottom of the aquifer.
- Note the following classifications of aquifers as unconfined, confined, or mixed (which start as confined prior to pumping and finish as unconfined after pumping is complete):
  - For unconfined aquifers, H > h0 (or specify as NA) in wellFlow and wellDrawdown, and H > h1 (or specify as NA) in kPump.
  - For confined aquifers, H >= hf in wellFlow and wellDrawdown, and H >= h2 in kPump.
  - For mixed aquifers, H < hf in wellFlow and wellDrawdown, and H >= hf in kPump.

Value

- `wellFlow` outputs the flow rate to the well (Q)
- `wellDrawdown` outputs a two-element list containing:
  1. h = height of groundwater surface a distance r from the well
  2. dd = h0 - h = drawdown of groundwater surface a distance r from the well
- `kPump` outputs the estimated hydraulic conductivity of the aquifer from the pumping test

Author(s)

James Kaklamanos <kaklamanosj@merrimack.edu> and Kyle Elmy <ElmyK@merrimack.edu>

See Also

`hydraulicConductivity`

Examples

```
##  Example code for Flow rate to well
wellFlow(k = 0.065, H = 10, h0 = 21, hf = 15, r0 = 20, rw = 2)

##  Example code for Well Drawdown
wellDrawdown(Q = 14.5, k = 0.065, H = 10, h0 = 21, r0 = 20, r = 2,
             rw = 2)

##  Example code for Hydraulic Conductivity from pumping tests
kPump(Q = 14.5, H = 10, h1 = 20, h2 = 16, r1 = 16, r2 = 8)
```

Index