Package: newmark (via r-universe)

May 19, 2026

Type Package

Title Uncertainty Analysis in Dynamic Site and Slope Response

Version 1.1.0

Description Implements a four-stage pipeline for probabilistic seismic performance analysis of slopes and embankments. The package takes a uniform-hazard spectrum at multiple return periods as input (any source) and produces: (1) synthetic soil profile generation and fundamental period estimation from USCS classification via Ishihara's small-strain shear-modulus model and the inhomogeneous truncated shear-beam theory of Gazetas and Dakoulas; (2) nonlinear site amplification using the Seyhan & Stewart (2014) model <doi:10.1193/063013EQS181M>, with inter-period correlation via Baker & Jayaram (2008) <doi:10.1193/1.2857544>; (3) Monte Carlo ensemble of six empirical Newmark sliding-block displacement models (Ambraseys & Menu (1988) <doi:10.1002/eqe.4290160704>, Jibson (2007) <doi:10.1016/j.enggeo.2007.01.013>, Saygili & Rathje (2008) <doi:10.1061/(ASCE)1090-0241(2008)134:6(790)>, Bray & Travasarou (2007) <doi:10.1061/(ASCE)1090-0241(2007)133:4(381)>, Bray & Macedo (2017) <doi:10.1016/j.soildyn.2017.05.024>, and the Bray and Macedo shallow-crustal update) with coherent correlated draws; (4) log-log inversion to the performance-based seismic coefficient kmax at user-specified displacement targets. All outputs are 'data.table' objects.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

LazyData true

LazyDataCompression xz

URL `https://github.com/averriK/newmark`,
`https://averriK.github.io/newmark/`

BugReports https://github.com/averriK/newmark/issues

RoxygenNote 7.3.2

Depends R (>= 4.2)

Imports data.table, digest, Hmisc, mvtnorm, randomForest, rpart, readxl, stats, stringr, triangle, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, devtools

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Config/roxygen2/version 8.0.0

Author Alejandro Verri Kozlowski [aut, cre, cph]

Maintainer Alejandro Verri Kozlowski <averri@fi.uba.ar>

Repository https://cran.r-universe.dev

Date/Publication 2026-05-19 08:42:04 UTC

RemoteUrl https://github.com/cran/newmark

RemoteRef HEAD

RemoteSha e140f74bec7d1e5b0496d4f0b155d3559d210dc2

Contents

approx.spline 3
buildGMDP 4
buildQSpline 5
checkUHS 5
CylinderRoots 6
designUHS 7
Dn_AM88 7
Dn_BM17 8
Dn_BM19 8
Dn_BT07 9
Dn_JB07 10
Dn_SR08 10
F_ST17 11
fitDnCurve 12
fitDnModel 13
fitModel.Ts 14
fitSaF 14
geSiteTable 15
getCylinderRoots 17
getDnKy 18
getKyLimits 19
getSiteProperties 19
interpolateSaTable 20

invertDnDraws . . . 21
rhoBJ . . . 22
SaF_ST17 . . . 23
sampleSaCorr . . . 24
ShearModelParameters . . . 25
SIDtoVs30 . . . 25
SiteClass . . . 26
SiteTable . . . 27
Vs30toSID . . . 28

Index 30

`approx.spline` *Highcharts-style cubic spline interpolation*

Description

Drop-in replacement for `stats::approx`. Uses the Catmull-Rom spline (with the same smoothing parameter as Highcharts `type = "spline"`) instead of straight-line segments. The call signature and return value are identical to `stats::approx`, so it can be substituted transparently.

Usage

```
approx.spline(
  x,
  y = NULL,
  xout = NULL,
  n = 50,
  rule = 1,
  log = FALSE,
  smoothing = 1.5,
  ...
)
```

Arguments

`x`	Numeric vector of abscissa values.
`y`	Numeric vector of ordinate values, same length as `x`.
`xout`	Optional numeric vector of points where the spline is evaluated. If `NULL`, a regular grid of length `n` is used.
`n`	Integer. Number of points to generate when `xout` is `NULL`. Default is 50.
`rule`	Handling of points outside the range of `x`. `1` returns `NA`; `2` repeats the nearest endpoint value. Matches the semantics of `stats::approx`.
`log`	Logical. If `TRUE`, interpolation is performed in log-log space. Default is `FALSE`.
`smoothing`	Numeric tension parameter used in the Catmull-Rom to Bezier conversion. Default is 1.5 (Highcharts default).
`...`	Additional arguments ignored; present only for full signature compatibility with `stats::approx`.

Value

A list with components x and y, just like stats::approx.

See Also

approx

Examples

```
x <- c(0.05, 0.10, 0.20, 0.30, 0.50, 1.0, 2.0, 3.0)
y <- exp(-x)
Td <- seq(0.05, 3.0, length.out = 100)
approx.spline(x, y, xout = Td, log = TRUE)$y
```

---

buildGMDP *Build Ground-Motion Design Parameters (GMDP)*

Description

Assembles an annual-exceedance-probability (AEP) table, uniform-hazard-spectrum (UHS) table, and–where available–a magnitude-distance disaggregation table, using OpenQuake or user-supplied hazard files.

Usage

```
buildGMDP(
  path,
  IDo = "GEM",
  engine = "openquake",
  vref = 760,
  TRo = c(100, 200, 500, 1000, 2000, 2500, 5000, 10000)
)
```

Arguments

path	Character. Directory with hazard data files.
IDo	Character. Identifier for the GMDP. Default "gmdp".
engine	Character. "openquake" (default) or "user".
vref	Numeric. Reference Vs30 (m/s). Default 760.
TRo	Numeric vector. Target return periods (years).

Value

A named list with AEPTable, UHSTable, and RMwTable (NULL if disaggregation is unavailable).

Examples

```
## Not run:
# Requires an 'OpenQuake' Engine output directory containing classical
# PSHA + disaggregation ZIPs; not bundled with the package.
out <- buildGMDP(path = "path/to/openquake/output", vref = 760)

## End(Not run)
```

`buildQSpline` *Build a monotone quantile spline Q(u) for ln(Sa)*

Description

Build a monotone quantile spline Q(u) for ln(Sa)

Usage

```
buildQSpline(SaTable)
```

Arguments

`SaTable`	data.table with probability column `p` (0-1 or "mean") and spectral-acceleration column `Sa`.

Value

A closure `Q(u)` returning `ln(Sa)` for any `u` `in[0,1]`.

Examples

```
tbl <- data.table::data.table(p = c(0.16, 0.50, 0.84), Sa = c(0.3, 0.5, 0.8))
Q <- buildQSpline(tbl)
Q(0.5)   # median ln(Sa)
```

`checkUHS` *Check monotonicity and duplicates in UHSTable*

Description

Ensures that, for every combination `TR`, `Vs30`, `Vref`, `ID`, `Tn`, there is exactly one row per probability `p` and that `Sa(p)` is non-decreasing.

Usage

```
checkUHS(UHSTable, epsMonot = 1e-06)
```

Arguments

`UHSTable`	data.table with columns `TR`, `Vs30`, `Vref`, `ID`, `Tn`, `p`, `Sa`.
`epsMonot`	numeric tolerance (default 1e-6).

Value

invisibly `TRUE`; stops on error.

Examples

```
## Not run:
# Requires a fully populated UHSTable produced by buildGMDP() or an
# equivalent upstream pipeline (must include the ID column that
# disaggregates per-quantile rows). Not bundled with the package.
checkUHS(UHSTable)

## End(Not run)
```

`CylinderRoots` *Cylinder Roots Data*

Description

This dataset contains cylinder roots.

Usage

```
data(CylinderRoots)
```

Format

A data frame with 307124 rows and 4 variables:

n Integer: Root number (1-10)

m Double: Inhomogeneity ratio (0-0.95)

l Double: Aspect Ratio (0-0.4950)

an Double: Cylinder roots

designUHS — *MCER / Design elastic spectrum for a Vs30/Vref slice (ASCE 7-22)*

Description

The input `UHS` slice must contain at least `Tn` and `SaF`. Typical usage passes a single TR and `p == "mean"`.

Usage

```
designUHS(UHS, TL = 8, spectrum = c("MCER", "DESIGN"))
```

Arguments

`UHS`	data.table slice from UHSTable (single TR, p == "mean").
`TL`	numeric, long-period transition (seconds). Default 8.
`spectrum`	character, "MCER" (default) or "DESIGN" (2/3 * MCER).

Value

data.table with columns `Tn, SaF`.

Examples

```
uhs <- data.table::data.table(Tn = c(0, 0.2, 0.5, 1, 2), SaF = c(0.4, 1.0, 0.9, 0.6, 0.3))
designUHS(uhs)
```

Dn_AM88 — *Ambraseys & Menu (1988) rigid sliding-block model*

Description

Ambraseys & Menu (1988) rigid sliding-block model

Usage

```
Dn_AM88(PGA, ky)
```

Arguments

`PGA`	Peak ground acceleration (g).
`ky`	Yield acceleration (g).

Value

data.table(muLnD, sdLnD, ID = "AM88")

Examples

```
Dn_AM88(PGA = 0.5, ky = 0.1)
```

`Dn_BM17` *Bray & Macedo (2017) - subduction/interface events*

Description

Bray & Macedo (2017) - subduction/interface events

Usage

```
Dn_BM17(ky, Sa, Ts, Mw = 7.5)
```

Arguments

`ky`	Yield acceleration (g).
`Sa`	Spectral acceleration at 1.5 * Ts (g).
`Ts`	Fundamental period (s).
`Mw`	Moment magnitude (default 7.5).

Value

data.table(muLnD, sdLnD, ID = "BM17")

Examples

```
Dn_BM17(ky = 0.1, Sa = 0.8, Ts = 0.3, Mw = 8.0)
```

`Dn_BM19` *Bray & Macedo (2019, corrected 2023) - shallow-crustal events*

Description

Multi-equation structure per BM23 (Soil Dynamics and Earthquake Engineering 168, 107835, DOI 10.1016/j.soildyn.2023.107835): Eq (10): ordinary motions, PGV <= 115 cm/s Eq (6): D100 max-component, PGV > 115 cm/s Eq (7): D50 median-component, PGV > 115 cm/s with a sub-regime split at PGV = 150 cm/s within Eqs (6) and (7).

Usage

```
Dn_BM19(ky, Ts, Sa, PGA, PGV, Mw = 6.5, NFC = "D100")
```

Arguments

ky	Yield acceleration (g).
Ts	Fundamental period (s).
Sa	Spectral acceleration at 1.3 * Ts (g).
PGA	Peak ground acceleration (g).
PGV	Peak ground velocity (cm/s).
Mw	Moment magnitude (default 6.5).
NFC	Near-fault component: "D100" (default; max-component, slope within 45 deg of fault-normal) or "D50" (median-component, other orientations). Used only when PGV > 115 cm/s.

Value

data.table(muLnD, sdLnD, ID = "BM19")

Examples

```
Dn_BM19(ky = 0.1, Ts = 0.3, Sa = 0.8, PGA = 0.5, PGV = 80, Mw = 7.0)
```

`Dn_BT07` *Bray & Travasarou (2007) flexible sliding-block model*

Description

Bray & Travasarou (2007) flexible sliding-block model

Usage

```
Dn_BT07(ky, Sa, Ts, Mw = 6.5)
```

Arguments

ky	Yield acceleration (g).
Sa	Spectral acceleration at 1.5 * Ts (g).
Ts	Fundamental period (s).
Mw	Moment magnitude.

Value

data.table(muLnD, sdLnD, ID = "BT07")

Examples

```
Dn_BT07(ky = 0.1, Sa = 0.8, Ts = 0.3, Mw = 7.0)
```

Dn_JB07 *Jibson (2007) empirical model*

Description

Jibson (2007) empirical model

Usage

```
Dn_JB07(PGA, ky, AI = NULL)
```

Arguments

PGA	Peak ground acceleration (g).
ky	Yield acceleration (g).
AI	Arias Intensity (m/s); if NULL, back-calculated from PGA.

Value

data.table(muLnD, sdLnD, ID = "JB07")

Examples

```
Dn_JB07(PGA = 0.5, ky = 0.1)
```

Dn_SR08 *Saygili & Rathje (2008) model*

Description

Saygili & Rathje (2008) model

Usage

```
Dn_SR08(PGA, ky, AI = NULL)
```

Arguments

PGA	Peak ground acceleration (g).
ky	Yield acceleration (g).
AI	Arias Intensity (m/s); if NULL, back-calculated from PGA.

Value

data.table(muLnD, sdLnD, ID = "SR08")

Examples

```
Dn_SR08(PGA = 0.5, ky = 0.1)
```

F_ST17 *Non-linear site amplification factor* **FST17**

Description

Implements the period-dependent, Vs30-dependent model of Seyhan & Stewart (2014) for the natural-log mean and standard deviation of the amplification factor $F(T_n, PGA, V_{s30})$.

Usage

```
F_ST17(PGA, Tn, vs30, vref = 760, Vl = 200, Vu = 2000)
```

Arguments

PGA	numeric **vector** – peak ground acceleration (in g).
Tn	numeric **scalar** – oscillator period (s).
vs30	numeric **scalar** – time-averaged shear-wave velocity to 30 m (m/s).
vref	numeric **scalar** – reference velocity (760 m/s or 3000 m/s), default 760.
Vl	numeric scalar – lower Vs30 bound for NL sigma interpolation (default 200).
Vu	numeric scalar – upper Vs30 bound for NL sigma interpolation (default 2000).

Details

If `vs30 == vref`*, the site factor equals 1 deterministically, so the function returns* `muLnF = 0` *and* `sdLnF = 0` *(zero dispersion;* `rnorm(n, 0, 0)` *returns zeros and* `ln F = 0`*).*

Value

A data.table with columns:

`muLnF` natural-log mean of F

`sdLnF` natural-log standard deviation of F (set to `0` when `vs30 == vref`)

`ID` character string `"ST17"`

References

Stewart, J.P. & Seyhan, E. (2017) “Semi-empirical nonlinear site amplification model for global application.” *Earthquake Spectra* **33**(1).

`fitDnCurve` *Newmark displacement curve Dn(ky) for one scenario*

Description

Evaluates ensemble and per-submodel Dn over a vector of ky values. The scenario is sampled once and the same epsilon is reused across all ky so that each realisation s represents a coherent physical curve Dn_s(ky).

Usage

```
fitDnCurve(uhs, ky, Ts, Mw = 6.5, NS = 30L, Rrup = 100, weights, NFC = "D100")
```

Arguments

`uhs`	data.table with columns Tn, p, Sa (one scenario)
`ky`	numeric vector, yield accelerations (g)
`Ts`	numeric scalar, fundamental period (s)
`Mw`	numeric scalar, moment magnitude
`NS`	integer, Monte Carlo samples for curve quantiles and per-realisation draws.
`Rrup`	numeric scalar, rupture distance (km)
`weights`	named numeric vector, ensemble weights by IDn
`NFC`	character, near-fault component selector for model BM19: `"D100"` (maximum component, default) or `"D50"`. Ignored by all other submodels.

Value

list(curve, draws). curve = data.table(ky, p, Dn, IDn, w). draws = data.table(ky, s, Dn, IDn) — always populated (possibly empty when no model produces output). Dn in cm.

Examples

```
## Not run:
uhs <- data.table::fread(
  system.file("extdata", "uhs.csv", package = "newmark"))
out <- fitDnCurve(uhs, ky = getDnKy(uhs, Ts = 0.3), Ts = 0.3, Mw = 7.5,
                  NS = 100, weights = c(AM88=1, BT07=1, SR08=1, BM17=1))

## End(Not run)
```

`fitDnModel` *Newmark displacement for a single submodel and scenario*

Description

Newmark displacement for a single submodel and scenario

Usage

```
fitDnModel(uhs, ky, Ts, Mw = 6.5, NS = 30L, Rrup = 100, IDn, NFC = "D100")
```

Arguments

`uhs`	data.table with columns Tn, p, Sa (one scenario)
`ky`	numeric scalar, yield acceleration (g)
`Ts`	numeric scalar, fundamental period (s)
`Mw`	numeric scalar, moment magnitude
`NS`	integer, Monte Carlo samples
`Rrup`	numeric scalar, rupture distance (km)
`IDn`	character, submodel identifier
`NFC`	character, near-fault component selector for model BM19: `"D100"` (maximum component, default) or `"D50"`. Ignored by all other submodels.

Value

data.table with columns p, Dn, IDn. Dn in cm.

Examples

```
## Not run:
uhs <- data.table::fread(
  system.file("extdata", "uhs.csv", package = "newmark"))
fitDnModel(uhs, ky = 0.1, Ts = 0.3, Mw = 7.5, NS = 100, IDn = "AM88")

## End(Not run)
```

fitModel.Ts *Site's Fundamental Period*

Description

Site's Fundamental Period

Usage

```
fitModel.Ts(VSm, hs, zm)
```

Arguments

VSm	Vector. Shear Wave Velocity profile in m/s (vector)
hs	Vector. Layer Thickness profile in m (vector)
zm	Vector. Layer Coordinates profile in m (vector)

Value

Double. Site's fundamental period Ts in seconds

fitSaF *Site-amplified AUXtral acceleration (Seyhan & Stewart 2014)*

Description

Computes site-amplified AUXtral acceleration using two modes: (A) If the input UHS contains only p == "mean", rock Sa is treated as deterministic and the dispersion in SaF comes solely from F_ST17 (sdLnF), sampled on the natural-log scale. (B) If the input UHS contains numeric quantiles (e.g., "0.10", "0.50", ...), ln Sa(Tn) is sampled from those quantiles using an empirical inverse quantile function (buildQSpline). Dependence between PGA and Sa(Tn) is represented by a Gaussian "star" copula (Baker & Jayaram, 2009) with correlation rhoBJ(Tn, T0 = 0, Rrup). For each Monte Carlo draw, F_ST17 is evaluated with the paired PGA draw, and ln SaF = ln Sa + ln F with ln F ~ Normal(muLnF, sdLnF). Independence between ln Sa and ln F is assumed.

Usage

```
fitSaF(
  uhs,
  vs30,
  vref = 760,
  ns = 1000,
  models = "ST17",
  Rrup = 100,
  p_TARGET = c(0.05, 0.1, 0.16, 0.5, 0.84, 0.9, 0.95)
)
```

Arguments

`uhs`	data.table with columns Tn, p, Sa.
`vs30`	target Vs30 (m/s).
`vref`	reference Vs30 (m/s), default 760.
`ns`	Monte Carlo size (default 1000).
`models`	character. Site-response model id. Currently the only implemented model is `"ST17"`; reserved for future site-response models.
`Rrup`	rupture distance (km) for rhoBJ() in the quantile case (default 100).
`p_TARGET`	probabilities to report when only 'mean' is provided, default c(0.05, 0.10, 0.16, 0.50, 0.84, 0.90, 0.95).

Details

Requirements:

- The input table must have columns Tn, p, Sa.
- It must include exactly one row (Tn == 0, p == "mean") providing PGA on rock.

Output columns: Tn, p, Sa, SaF, AF, where AF = SaF / Sa. In mode (B), the same numeric p values found in the input are returned plus "mean". In mode (A), the set of p values is taken from p_TARGET plus "mean".

Value

data.table with columns Tn, p, Sa, SaF, AF.

Examples

```
## Not run:
uhs <- data.table::fread(
  system.file("extdata", "uhs.csv", package = "newmark"))
fitSaF(uhs, vs30 = 360, vref = 760, ns = 500)

## End(Not run)
```

`geSiteTable` *Single-realisation synthetic soil profile and site properties*

Description

Builds one Monte Carlo realisation of a layered soil column from a total height `Hs` and a USCS classification, samples random geotechnical properties (void ratio, unit weight, plasticity, shear modulus parameters), computes the fundamental period `Ts` via Rayleigh's method (`fitModel.Ts`) and the Vs30-equivalent shear-wave velocity. `getSiteProperties` calls this function repeatedly to assemble the Monte Carlo ensemble; call directly when one realisation suffices.

Usage

```
geSiteTable(
  Hs,
  Water = 0,
  USCS,
  Group = NULL,
  h = 0.5,
  DrID = NULL,
  Vref = 760,
  UniformDistribution = TRUE,
  POP = 100,
  IgnoreModelIntervals = TRUE,
  getSiteLayers = FALSE
)
```

Arguments

- `Hs` Numeric scalar. Total height to hard ground (m).
- `Water` Numeric scalar. Water-table depth as fraction of `Hs`: `zw = Hs - Water * Hs`.
- `USCS` Character vector. Unified Soil Classification System codes (e.g. `c("GC", "CL", "ML")`).
- `Group` Character vector or `NULL`. Soil groups, `c("Gravels","Fines","Sands")`; if `NULL`, derived from `USCS`.
- `h` Numeric scalar. Layer thickness in the discretisation (m). Default 0.50.
- `DrID` Character or `NULL`. Relative-density label, `c("Very Dense","Dense","Compact","Loose","Very Loose")`; `NULL` samples relative density from the standard prior.
- `Vref` Numeric. Reference shear-wave velocity (m/s) for the site-class assignment. Default 760.
- `UniformDistribution` Logical. If `TRUE` (default), sample soil properties from uniform distributions; if `FALSE`, use normal distributions.
- `POP` Numeric. Pre-consolidation pressure (kPa); enters the over-consolidation ratio `OCR = (pm + POP) / pm`. Default 100.
- `IgnoreModelIntervals` Logical. If `TRUE` (default), ignore model-specific layer-thickness intervals and use `h` as the discretisation step.
- `getSiteLayers` Logical. If `TRUE`, return the per-layer detail in addition to the per-site summary.

Value

data.table with one row per (Hs, POP, Hw, sample) combination and columns Hs, Hw, NL, Z500, Z1000, SID, Go, mo, Ts, VSo, VS30, plus USCS-fraction columns.

`getCylinderRoots` *Characteristic root of an inhomogeneous truncated shear beam*

Description

Returns the eigenvalue `an` of mode `no` for a Gazetas & Dakoulas (1985) inhomogeneous truncated shear beam parameterised by inhomogeneity ratio `mo` and truncation ratio `lo`. The value is interpolated from the precomputed `CylinderRoots` table using the requested model (linear glm, non-linear glm, decision tree, or random forest).

Usage

```
getCylinderRoots(mo, lo, no = 1, model = "nlm", extrapolate = TRUE, OSF = 0.1)
```

Arguments

`mo`	Double. Inhomogeneity ratio of the shear-modulus profile. `mo = 0` is a homogeneous column. Range 0 to 0.95.
`lo`	Double. Truncation ratio (top-radius / bottom-radius); `lo = 0` is a fully truncated cone. Range 0 to 0.495.
`no`	Integer. Mode number, 1..8. Default 1 (fundamental).
`model`	Character. Interpolation model: `"lm"` (linear GLM), `"nlm"` (non-linear GLM with interactions, default), `"dt"` (decision tree), or `"rf"` (random forest).
`extrapolate`	Logical. If `TRUE` (default), allow `mo`/`lo` outside the table support and warn; if `FALSE`, clamp to the table maxima.
`OSF`	Double. Outlier scale factor used by the random-forest model to shrink the training set around the query point. Default 0.10.

Value

Double. Characteristic root `an` for the requested mode and parameters.

Examples

```
getCylinderRoots(mo = 0.45, lo = 0.20, no = 1, model = "nlm")
```

`getDnKy` *Build a log-spaced ky grid for one scenario*

Description

Derives the effective intensities PGA, Sa(1.3Ts), Sa(1.5Ts) from the scenario UHS and constructs a log-spaced ky grid covering the calibrated range of the displacement models.

Usage

```
getDnKy(uhs, Ts, kyN = 30L, pRef = "mean")
```

Arguments

`uhs`	data.table with columns Tn, p, Sa (one scenario)
`Ts`	numeric scalar, fundamental period (s)
`kyN`	integer, number of grid points (default 30)
`pRef`	character, probability level for intensity anchor (default "mean")

Details

Default bounds come from the calibration ranges of the models in the ensemble:

- lower: 0.01g (BM17 lowest calibration point)
- upper: max(PGA, 0.80g). The physical limit is ky < PGA (no slip below the yield acceleration); the 0.80g floor extends the grid for low-PGA scenarios up to BM17's calibration ceiling.

Value

numeric vector of ky values (log-spaced)

Examples

```
uhs <- data.table::data.table(Tn = 0, p = "mean", Sa = 0.5)
getDnKy(uhs, Ts = 0.3)
```

getKyLimits — *Get calibrated ky range for each displacement model*

Description

Returns the ky bounds within which each model's regression is calibrated. For rigid models (AM88, SR08) the bounds are relative to PGA. For flexible models the bounds are absolute.

Usage

```
getKyLimits(PGA)
```

Arguments

`PGA`	numeric scalar, peak ground acceleration (g). Required for rigid model limits.

Value

data.table with columns IDn, ky.lo, ky.hi

Examples

```
getKyLimits(PGA = 0.5)
```

getSiteProperties — *Monte Carlo site properties from synthetic soil profiles*

Description

Generates `NR` realisations of synthetic soil profiles for a given total height `Hs` and USCS classification, then summarises the resulting Ts, mo, Go, VSo and VS30 by user-requested quantile levels (and/or the mean). Each realisation is built by `geSiteTable()`.

Usage

```
getSiteProperties(
  Hs,
  USCS,
  POP = 100,
  Water = 0,
  NR = 1,
  h = 1,
  levels = c(0.05, 0.5, "mean", 0.95),
  Vref = 760
)
```

Arguments

Hs	Numeric vector. Total height(s) to hard ground (m).
USCS	Character vector. USCS soil classification codes (e.g. `c("GC", "CL", "ML")`).
POP	Numeric. Pre-consolidation pressure (kPa). Default 100.
Water	Numeric. Water-table depth as fraction of `Hs`. Default 0 (no water).
NR	Integer. Number of Monte Carlo realisations. Default 1.
h	Numeric. Layer thickness (m) in the discretisation. Default 1.00.
levels	Vector. Quantile levels to report; entries can be numeric in (0, 1) and/or the literal `"mean"`. Default `c(0.05, 0.5, "mean", 0.95)`.
Vref	Numeric. Reference shear-wave velocity (m/s) for site-class assignment. Default 760.

Value

data.table with one row per (Hs, POP, Hw, level) combination and columns USCS, Go, mo, Ts, VSo, VS30, Z500, Z1000, level. Numeric properties are rounded for readability.

Examples

```
getSiteProperties(
  Hs   = 30,
  USCS = c("GC", "CL", "ML"),
  NR   = 50,
  levels = c(0.16, "mean", 0.84)
)
```

---

interpolateSaTable *Interpolate Sa(p) at arbitrary period using log-log interpolation*

Description

Interpolate Sa(p) at arbitrary period using log-log interpolation

Usage

```
interpolateSaTable(uhs, Tn)
```

Arguments

uhs	data.table with columns Tn, p, Sa.
Tn	numeric scalar — target period for interpolation.

Value

data.table with interpolated Tn, p, Sa

Examples

```
uhs <- data.table::data.table(
  Tn = c(0.2, 0.5, 1.0, 0.2, 0.5, 1.0),
  p  = c("0.16", "0.16", "0.16", "0.84", "0.84", "0.84"),
  Sa = c(0.8, 0.6, 0.3, 1.2, 0.9, 0.5)
)
interpolateSaTable(uhs, Tn = 0.3)
```

`invertDnDraws` *Invert Dn draws to kmax(Da)*

Description

For each realisation s, inverts each model's Dn(ky) curve to kmax(Da), builds weighted ensemble mean, then averages over realisations.

Usage

```
invertDnDraws(draws, Da, weights, p = c(0.05, 0.1, 0.16, 0.84, 0.9, 0.95))
```

Arguments

`draws`	data.table(ky, s, Dn, IDn) — Dn in cm
`Da`	numeric vector — displacement targets in cm
`weights`	named numeric vector, ensemble weights by IDn
`p`	numeric quantiles in (0,1) to report alongside the mean. Default c(0.05, 0.10, 0.16, 0.84, 0.90, 0.95). The output always includes the ensemble mean as the first p row.

Details

Log-log linear extrapolation is used when Da falls outside the support of the computed Dn(ky) curve. No calibration-range clamping is applied — the natural support of the curve bounds the inversion.

All inputs must be in the same displacement units (cm).

Value

data.table(Da, p, kmax). Column `p` is character: "mean" or the numeric quantile formatted as "0.84", "0.90", etc.

Examples

```
## Not run:
uhs <- data.table::fread(
  system.file("extdata", "uhs.csv", package = "newmark"))
out <- fitDnCurve(uhs, ky = getDnKy(uhs, Ts = 0.3), Ts = 0.3, Mw = 7.5,
                  NS = 100,
                  weights = c(AM88=1, BT07=1, SR08=1, BM17=1))
invertDnDraws(out$draws, Da = c(2.5, 5, 10),
              weights = c(AM88=1, BT07=1, SR08=1, BM17=1))

## End(Not run)
```

`rhoBJ` *Baker and Jayaram (2008) inter-period correlation model*

Description

Computes the Pearson correlation coefficient between the residuals of ln(Sa) at two oscillator periods, using the canonical predictive equation of Baker and Jayaram (2008), Earthquake Spectra 24(1), 299-317 (DOI 10.1193/1.2857544), Eqs. 5 and 6 verbatim.

Usage

```
rhoBJ(Tn, T0 = 0.01, Rrup = 100)
```

Arguments

`Tn`	Numeric scalar. Target spectral period in seconds.
`T0`	Numeric scalar. Reference period in seconds. Default 0.01 (used as a numerical proxy for PGA). Values <= 0 are silently converted to 0.01.
`Rrup`	Numeric scalar. Not used; retained for backward compatibility.

Details

The model has no dependence on rupture distance or magnitude. `Rrup` is retained for signature compatibility and is silently ignored. Periods of zero are treated as 0.01 s (standard PGA proxy in B&J 2008).

Form (Eqs. 5-6 of B&J 2008): C1 = 1 - cos(pi/2 - 0.366 * ln(Tmax / max(Tmin, 0.109))) C2 = (1 - 0.105 * (1 - 1/(1 + exp(100*Tmax - 5))) * (Tmax-Tmin)/(Tmax-0.0099)) if Tmax < 0.2; else 0 C3 = C2 if Tmax < 0.109; else C1 C4 = C1 + 0.5 * (sqrt(C3) - C3) * (1 + cos(pi * Tmin / 0.109))

rho = C2 if Tmax < 0.109 = C1 else if Tmin > 0.109 = min(C2, C4) else if Tmax < 0.2 = C4 otherwise

Validity: 0.01 <= T1, T2 <= 10 s.

Value

Numeric scalar correlation coefficient rho(T0, Tn).

References

Baker, J. W. & Jayaram, N. (2008). Correlation of Spectral Acceleration Values from NGA Ground Motion Models. Earthquake Spectra, 24(1), 299-317. DOI: 10.1193/1.2857544.

Examples

```
rhoBJ(Tn = 0.3, T0 = 0.01)   # rho(PGA, 0.3 s)
rhoBJ(Tn = 1.0, T0 = 0.5)
```

SaF_ST17 *Site-amplified spectral acceleration using Seyhan & Stewart (2014)*

Description

Computes site-amplified spectral acceleration Sa_F using the Seyhan & Stewart (2014) non-linear site factor model. This is the canonical implementation that calls the underlying F_ST17() function.

Usage

```
SaF_ST17(Sa, pga, Tn, vs30, vref = 760)
```

Arguments

`Sa`	numeric vector - rock spectral accelerations at period Tn (g)
`pga`	numeric vector - peak ground acceleration on rock (g), same length as Sa
`Tn`	numeric scalar - oscillator period (s)
`vs30`	numeric scalar - time-averaged shear-wave velocity to 30 m (m/s)
`vref`	numeric scalar - reference velocity (m/s), default 760

Value

A data.table with columns:

`muLnSaF` natural-log mean of amplified Sa

`sdLnSaF` natural-log standard deviation of amplified Sa

`ID` character string "ST17"

References

Seyhan, E. & Stewart, J.P. (2014) "Semi-empirical nonlinear site amplification from NGA-West2 data and simulations." *Earthquake Spectra* **30**(3):1241–1256. doi:10.1193/063013EQS181M

See Also

`F_ST17` for the underlying site factor model

`fitSaF` for Monte-Carlo site amplification analysis

Examples

```
library(data.table)

# Site amplification for soft soil site
Sa_rock <- c(0.1, 0.2, 0.3, 0.4)
PGA_rock <- c(0.1, 0.1, 0.1, 0.1)

result <- SaF_ST17(
  Sa = Sa_rock,
  pga = PGA_rock,
  Tn = 0.5,      # 0.5s period
  vs30 = 300,    # Soft soil
  vref = 760     # Rock reference
)

print(result)
```

---

sampleSaCorr — *Draw a correlated sample of spectral accelerations*

Description

Draw a correlated sample of spectral accelerations

Usage

```
sampleSaCorr(uhs, Tn, rho, NS)
```

Arguments

uhs	data.table with columns Tn, p, Sa (must include p == "mean").
Tn	numeric vector; the first element is the reference period `Tn[1]`, e.g., 0.01 or 0.00. It does not have to be sorted.
rho	numeric vector of length `length(Tn) - 1`; correlations between the reference period `Tn[1]` and each remaining period.
NS	integer, Monte Carlo sample size.

Value

matrix of dimension NS x length(Tn) with Sa draws (g).

Examples

```
## Not run:
uhs <- data.table::data.table(
  Tn = c(0, 0, 0, 1, 1, 1),
  p  = c("0.16", "mean", "0.84", "0.16", "mean", "0.84"),
  Sa = c(0.3, 0.5, 0.7, 0.1, 0.2, 0.3)
```

```
)
rho <- rhoBJ(Tn = 1.0, T0 = 0.01)
mat <- sampleSaCorr(uhs, Tn = c(0, 1), rho = rho, NS = 100)

## End(Not run)
```

`ShearModelParameters` *ShearModelParameters*

Description

This dataset contains hear Model Parameters from Ishihara (1996)

Usage

```
data(ShearModelParameters)
```

Format

A data frame with 20 rows and 10 variables:

ModelID Character: Model ID

GroupID Character: Group ID

NameID Character: NameID

AuthorID Character: AuthorID

emin Double: Min Void Ratio

emax Double: Max Void Ratio

A Double: A

Ce Double: Ce

n Double: Ce

UN Character: Go Units

`SIDtoVs30` *Convert Site Class to Vs30 in m/s*

Description

Convert Site Class to Vs30 in m/s

Usage

```
SIDtoVs30(SID = NULL)
```

Arguments

`SID`	Site Class

Value

Vs30 in m/s

Examples

```
SIDtoVs30("A")

SIDtoVs30("AB")

SIDtoVs30(c("BC","C","CD","D"))
```

`SiteClass` *Site Class Data*

Description

This dataset contains Site Class data from ASCE 7-22

Usage

```
data(SiteClass)
```

Format

A data frame with 9 rows and 4 variables:

SC Character: Site Class

Description Character: Category description

Vs30 (m/s) Character: Vs30 in m/s

Vs30 (ft/s) Character: Vs30 in ft/s

SiteTable *SiteTable Data*

Description

This dataset contains simulations of different site conditions

Usage

```
data(SiteTable)
```

Format

A data frame with 255614 rows and 38 variables:

Hs Double: Height in m.

Hw Double: Water Table in m.

NL Integer: Number of Layers

Z500 Double: Depth to 500 m/s in m.

Z1000 Double: Depth to 1000 m/s in m.

SID String: Site ID

Go Double: Shear Modulus in kPa.

mo Double: Shear Modulus Exponent.

Ts Double: Fundamental Period in s.

VSo Double: Shear Wave Velocity in m/s.

VS30 Double: Shear Wave Velocity at 30 m in m/s.

UID String: USCS ID

Gravels Double: Percentage of Gravels.

Sands Double: Percentage of Sands.

Fines Double: Percentage of Fines.

Clays Double: Percentage of Clays.

Silts Double: Percentage of Silts.

Organic Double: Percentage of Organic.

Water Double: Percentage of Water.

GP Double: GP Fraction.

GM Double: GM Fraction.

GC Double: GC Fraction.

GW Double: GW Fraction.

SP Double: SP Fraction.

SM Double: SM Fraction.

SC Double: SC Fraction.

SW Double: SW Fraction.

CL Double: CL Fraction.

ML Double: ML Fraction.

OL Double: OL Fraction.

PT Double: PT Fraction.

OH Double: OH Fraction.

MH Double: MH Fraction.

CH Double: CH Fraction.

POP Double: Overpressure in Kpa.

POP_Units String: Overpressure Units.

Go_Units String: Shear Modulus Units.

Vs_Units String: Shear Wave Velocity Units.

`Vs30toSID` *Convert Vs30 (m/s) to Site Class Identifier*

Description

Given one or more Vs30 values (in m/s), returns a character vector of site class designations (`"A"`, `"B"`, `"BC"`, `"C"`, `"CD"`, `"D"`, `"DE"`, `"E"`). The classification thresholds are:

- A: Vs30 >= 1500
- B: 900 <= Vs30 < 1500
- BC: 640 <= Vs30 < 900
- C: 440 <= Vs30 < 640
- CD: 300 <= Vs30 < 440
- D: 210 <= Vs30 < 300
- DE: 150 <= Vs30 < 210
- E: 0 <= Vs30 < 150

Usage

```
Vs30toSID(Vs30)
```

Arguments

`Vs30`	Numeric vector. One or more Vs30 values in m/s.

Value

A character vector of the same length as `Vs30`, with site class designations.

Examples

```
Vs30toSID(1500)  # returns "A"
Vs30toSID(c(120, 790, 3000, 455))
```

Index

∗ **datasets**
CylinderRoots, 6
ShearModelParameters, 25
SiteClass, 26
SiteTable, 27

approx, *4*
approx.spline, 3

buildGMDP, 4
buildQSpline, 5

checkUHS, 5
CylinderRoots, 6

data.table, *11*, *23*
designUHS, 7
Dn_AM88, 7
Dn_BM17, 8
Dn_BM19, 8
Dn_BT07, 9
Dn_JB07, 10
Dn_SR08, 10

F_ST17, 11, *23*
fitDnCurve, 12
fitDnModel, 13
fitModel.Ts, 14
fitSaF, 14, *23*

geSiteTable, 15
getCylinderRoots, 17
getDnKy, 18
getKyLimits, 19
getSiteProperties, 19

interpolateSaTable, 20
invertDnDraws, 21

rhoBJ, 22

SaF_ST17, 23
sampleSaCorr, 24
ShearModelParameters, 25
SIDtoVs30, 25
SiteClass, 26
SiteTable, 27

Vs30toSID, 28