

Package: calculiva (via r-universe)

October 24, 2026

Type Package

Title Home Improvement Estimators for Mulch, Topsoil and Sod

Version 0.3.0

Description Planning quantities for home improvement projects: mulch by bed, tree ring, known area, play structure or swing; topsoil and fill dirt by area with bags, bulk, weight and trips; sod by area with a staggered piece layout, pallets, plugs and soil preparation. A port of the Go module 'calculiva-go' (same version and test vectors), itself a port of the engine of the calculators on <<https://calculiva.com/>>. The method is described at <<https://calculiva.com/how-we-calculate/>>.

License MIT + file LICENSE

URL <https://calculiva.com/>, <https://calculiva.com/how-we-calculate/>,
<https://github.com/calculiva/calculiva-r>

BugReports <https://github.com/calculiva/calculiva-r/issues>

Encoding UTF-8

Depends R (>= 3.5.0)

Suggests jsonlite, testthat (>= 3.0.0)

Config/testthat/edition 3

Repository <https://tresor4k.r-universe.dev>

Date/Publication 2026-10-24 11:19:26 UTC

RemoteUrl <https://github.com/calculiva/calculiva-r>

RemoteRef HEAD

RemoteSha c5584e850634b44b59078ba875d92638bc337fb4

Contents

calculiva-package	2
calculiva_units	2
mulch_new_plan	3
sod_new_plan	4
soil_new_plan	5

Index

7

calculiva-package	<i>Calculiva home improvement estimators</i>
-------------------	--

Description

Planning quantities for mulch, topsoil and fill, and sod: volume, bags, bulk, pallets, pieces, weight and trips. A port of the Go module github.com/calculiva/calculiva-go (same version, same test vectors), itself a port of the engine of the calculators on <https://calculiva.com/>. The method is described at <https://calculiva.com/how-we-calculate/>.

All lengths are millimetres (see [calculiva_units](#)). Inputs and results are named lists whose names follow the Go module's fields in lower camel case (areaMM2, cuFt, bagCuFt); a NULL value means that the input it depends on was not entered. Invalid input signals a condition of class `calculiva_input_error` whose field names the offending field and whose `input_message` is the message alone.

Every constant carries, in the source code, its data sheet, the public source it was read on and the date it was verified. Material weights, bag and wheelbarrow sizes and sod allowances are not shipped: they are inputs of the user. The results are planning quantities: the product label, the supplier's quote, the local code and the installer on site govern.

See Also

[mulch_new_plan](#), [soil_new_plan](#), [sod_new_plan](#).

Examples

```
ft <- calculiva_units()$foot
inch <- calculiva_units()$inch
# Two 40 x 3 ft beds topped up from 1 in to 3 in, bought in 2 cu ft bags.
bed <- list(id = "front-bed", kind = list(type = "bed", l = 40 * ft, w = 3 * ft, count = 2),
           target = 3 * inch, existing = 1 * inch)
plan <- mulch_new_plan(list(areas = list(bed), bagSize = 2))
plan$totals$cuFt
plan$buy$bags
```

calculiva_units	<i>Units and sourced constants</i>
-----------------	------------------------------------

Description

The length units and the sourced constants of each estimator, as named lists.

Usage

```
calculiva_units()
soil_constants()
mulch_constants()
sod_constants()
```

Value

calculiva_units returns inch (25.4) and foot (304.8), in millimetres. soil_constants returns the NIST Handbook 130 units (pounds per short ton, cubic feet per cubic yard, cubic inches per dry quart) and the square and cubic foot in mm. mulch_constants returns the CPSC playground surfacing rules and the mulch depth guidance publisher by publisher. sod_constants returns the surveyed sod formats and pallets, the Penn State and NC State soil preparation figures and UF/IFAS LH013 Table 3. The source and verification date of each value are given in the package source.

Examples

```
calculiva_units()$foot
soil_constants()$cubicFeetPerCubicYard
```

mulch_new_plan	<i>Mulch by area</i>
----------------	----------------------

Description

Volume per bed, tree ring, known area, play structure or swing (CPSC depth and extent), bags, bulk, break-even, loads and warnings. Calculator page: <https://calculiva.com/landscaping/mulch-calculator/>.

Usage

```
mulch_new_rules()
mulch_new_ring_guidance()
mulch_check_ring(r, gap, guidance = mulch_new_ring_guidance())
mulch_new_plan(input, rules = mulch_new_rules())
mulch_coverage(bag_cu_fts, depths_in, rules = mulch_new_rules())
```

Arguments

r	the ring radius, mm.
gap	the bare gap at the trunk, mm.
guidance	the published gaps and radii from mulch_new_ring_guidance.
input	a list with areas (one to eight lists with id, kind, target and existing depths in mm), texture ("coarse" or "fine"), bagSize, bagUnit ("cuFt" or "liters"), bagPrice, bulkPrice (per cu yd), delivery, increment (cu yd), extra (per cent), bedVolume and wheelbarrow (cu ft). A kind is a list with type "bed" (1,

w, count), "ring" (r, trunk, gap, count), "knownArea" (area, mm2), "play" (l, w, height) or "swing" (beam, height). Missing fields take the calculator default.

rules the rules from mulch_new_rules.

bag_cu_fts bag volumes, cubic feet.

depths_in depths, inches.

Value

mulch_new_plan returns a list with areas, totals, buy, trips, wheelbarrowLoads and warnings; mulch_check_ring the published minimums with ok for each; mulch_coverage the square feet per bag at each depth and the bags per cubic yard.

Examples

```
ft <- calculiva_units()$foot
swing <- list(id = "s1", kind = list(type = "swing", beam = 10 * ft, height = 8 * ft))
mulch_new_plan(list(areas = list(swing)))$totals$cuYd
mulch_check_ring(3 * ft, 76.2)$radii[[1]]$ok
```

sod_new_plan	<i>Sod by area</i>
--------------	--------------------

Description

Pieces per area (rectangles laid out piece by piece in staggered rows, with offcut pairing), pallets, plugs (UF/IFAS LH013 Table 3), soil preparation, cost and weight. Calculator page: <https://calculiva.com/landscaping/sod-calculator/>.

Usage

```
sod_new_rules()
sod_new_input(areas = list(), rules = sod_new_rules())
sod_new_plan(input, rules = sod_new_rules())
sod_lay_rect(l, w, piece_w, piece_l)
sod_format_table(sold_as, sqft, rules = sod_new_rules())
```

Arguments

areas one to eight lists with id and shape: type "rect" (l, w, count), "circle" (r, count), "triangle" (b, h, count) or "knownArea" (area, mm2); lengths in mm.

rules the rules from sod_new_rules; set allowance and allowanceComplex (percent) yourself.

input	the list returned by sod_new_input, with palletPieces (or palletMode = "sqFt" and palletSqFt), format (a format id or "custom" with pieceW and pieceL), allowance, spare, method ("solid" or "plugs"), species, spacing, plugSize, topsoil, organic ("layer", "rate" or "none"), organicIn, organicRate, priceMode ("perPallet", "perPiece" or "perSqFt"), price, delivery and payload.
l, w	the rectangle length and width, mm.
piece_w, piece_l	the piece width and length, mm.
sold_as	pallet coverages, square feet.
sqft	the lawn area, square feet.

Value

sod_new_plan returns a list with the piece size, areas, totals, pallets, plug, prep, cost, weightLb, trips and warnings; sod_lay_rect the placed pieces, rows and counts; sod_format_table one row per format.

Examples

```
ft <- calculiva_units()$foot
lawn <- list(id = "a1", shape = list(type = "rect", l = 10 * ft, w = 8 * ft, count = 1))
input <- sod_new_input(list(lawn))
input$palletPieces <- 150
sod_new_plan(input)$totals$pieces
```

soil_new_plan	<i>Topsoil and fill dirt by area</i>
---------------	--------------------------------------

Description

Volume per area, bags, bulk by the yard or the ton, weight, truck trips and wheelbarrow loads, per material (topsoil first, then fill). Calculator page: <https://calculiva.com/landscaping/soil-calculator/>.

Usage

```
soil_new_rules(weights = list())
soil_area_of(id, shape)
soil_bag_cu_ft(size, unit, dry_quart_in3)
soil_new_plan(input, rules)
soil_coverage(depths_in, rules, weight_id)
```

Arguments

weights	a list of material weights entered by the user, each a list with id, label, loose and bank (pounds per cubic yard) and loadFactor.
id	the area id; it prefixes the field of any error ("a1-1").
shape	a list with type "rect" (l,w), "circle" (d), "triangle" (b,h) or "knownArea" (area, mm2); lengths in mm.
size	the bag size.
unit	"cuFt", "dryQuarts" or "liters".
dry_quart_in3	cubic inches per dry quart, from the rules.
input	a list with areas (one to eight lists with id, shape, depth in mm, count and material "topsoil" or "fill"), topsoil and fill (lists with bagSize, bagUnit, bagPrice, bulkPrice and the weight id), bulkBy ("yard" or "ton"), delivery, increment (cu yd), extra (percent), payload (lb), bedVolume and wheelbarrow (cu ft). Missing fields take the calculator default (0, "yard").
rules	the rules from soil_new_rules.
depths_in	depths in inches.
weight_id	the id of a weight of the rules.

Value

soil_new_plan returns a list with areas, materials, totals and warnings; soil_area_of an area in mm2; soil_bag_cu_ft a volume in cubic feet; soil_coverage one row per depth (square feet per cubic yard, per ton and per cubic foot).

Examples

```
ft <- calculiva_units()$foot
rules <- soil_new_rules(list(list(id = "mine", label = "my supplier's ticket",
                                loose = 2000, bank = 2500, loadFactor = 0.75)))
area <- list(id = "a1", shape = list(type = "rect", l = 10 * ft, w = 12 * ft),
            depth = 101.6, count = 1, material = "topsoil")
topsoil <- list(bagSize = 0.75, weight = "mine")
plan <- soil_new_plan(list(areas = list(area), topsoil = topsoil), rules)
plan$materials[[1]]$bags
```

Index

* **package**

calculiva-package, [2](#)

calculiva (calculiva-package), [2](#)

calculiva-package, [2](#)

calculiva_units, [2](#), [2](#)

mulch_check_ring (mulch_new_plan), [3](#)

mulch_constants (calculiva_units), [2](#)

mulch_coverage (mulch_new_plan), [3](#)

mulch_new_plan, [2](#), [3](#)

mulch_new_ring_guidance
(mulch_new_plan), [3](#)

mulch_new_rules (mulch_new_plan), [3](#)

sod_constants (calculiva_units), [2](#)

sod_format_table (sod_new_plan), [4](#)

sod_lay_rect (sod_new_plan), [4](#)

sod_new_input (sod_new_plan), [4](#)

sod_new_plan, [2](#), [4](#)

sod_new_rules (sod_new_plan), [4](#)

soil_area_of (soil_new_plan), [5](#)

soil_bag_cu_ft (soil_new_plan), [5](#)

soil_constants (calculiva_units), [2](#)

soil_coverage (soil_new_plan), [5](#)

soil_new_plan, [2](#), [5](#)

soil_new_rules (soil_new_plan), [5](#)