Package ‘mknapsack’

April 10, 2018

Type Package

Title Multiple Knapsack Problem Solver

Version 0.1.0

Description Package solves multiple knapsack optimisation problem. Given a set of items, each with volume and value, it will allocate them to knapsacks of a given size in a way that value of top N knapsacks is as large as possible.

License GPL-2

URL https://github.com/madedotcom/mknapsack

BugReports https://github.com/madedotcom/mknapsack/issues

Encoding UTF-8

LazyData true

Suggests testthat, mockery, Rglpk, ROI, ROI.plugin.glpk

Imports assertthat, data.table, lpSolve

RoxygenNote 6.0.1

NeedsCompilation no

Author Bulat Yapparov [aut, cre],
MADE.com [cph]

Maintainer Bulat Yapparov <bulat.yapparov@made.com>

Repository CRAN

Date/Publication 2018-04-10 12:45:53 UTC

R topics documented:

  group_moq .............................................................. 2
  knapsack ............................................................... 2
  mknapsack ............................................................. 3
  moq_constraint ....................................................... 4
  unitsbro .............................................................. 4

Index 5
### group_moq

**Collapse function for the MOQ items**

**Description**

Combines items with MOQ greater than one to a single line that represents min amount that can be ordered.

**Usage**

```
group_moq(units)
```

**Arguments**

- `units` data.table with following fields: sku, utility, volume, moq

**Value**

data.table with sku, utility, volume and units fields. first lines for each sku are grouped according to moq

### knapsack

**Solves knapsack problem with the library defined in knapsack.solver option:**
- cbc (default)
- uses rcbc package
- lpsolve - uses lpSolve package

**Description**

Solves knapsack problem with the library defined in knapsack.solver option: - cbc (default) - uses rcbc package - lpsolve - uses lpSolve package

**Usage**

```
knapsack(profit, volume, moq = rep(0, length(profit)), cap = 65)
```

**Arguments**

- `profit` vector with profit for item
- `volume` vector of item sizes in cubic meters
- `moq` vector of flags where 1 means that row contains minimum order quantity (MOQ). Defaults to zero vector matching profit in length.
- `cap` size of the container in cubic meters

**Value**

vector with container numbers keeping the permutation of the original data
mknapsack

Optimal packing into multiple containers

Description

Gets containers based on the utility of individual items, their volume and container size.

Usage

mknapsack(profit, volume, moq = rep(0, length(profit)), cap = 65,
          sold = rep(0, length(profit)))

Arguments

profit           vector with profit for item
volume           vector of item sizes in cubic meters
moq             vector of flags where 1 means that row contains minimum order quantity (MOQ). Defaults to zero vector matching profit in length.
cap             size of the container in cubic meters
sold           vector with a number of items that were sold on demand

Value

vector with container numbers keeping the permutation of the original data

Examples

# Calculate the optimal containers summary for a sample dataset
data(unitsbro)
library(data.table)
units.combined <- data.table(unitsbro)
moq <- units.combined$moq
profit <- units.combined$utility
volume <- units.combined$volume
res <- mknapsack(profit, volume, moq, 65)
units.combined$container <- as.factor(res)
#Aggregate solution to container
containers <- units.combined[order(container), .(volume = sum(volume),
profit = sum(profit)), by = container]
moq_constraint  

**Description**

Creates matrix of moq constraints for the LP optimisation. It is assumed that there is only one moq position per SKU and data is sorted by sku, therefore SKU index can be calculated.

**Usage**

```r
moq_constraint(moq)
```

**Arguments**

- `moq`: flag that indicates that this position contains MOQ

**Value**

Matrix that expresses the MOQ constraint: non-MOQ item cannot be put into container that does not contain MOQ item.

---

unitsbro  

**Description**

Dataset contains line items with utility and volume and can be used for exploration of the package functionality.

**Usage**

```r
unitsbro
```

**Format**

A data frame with rows and variables

- `sku`: identifier for the product
- `utility`: proxy of the profit that this item delivers to the company if purchased
- `volume`: volume of the item, usually in cubic meters
- `units`: number of units that this line contains
- `moq`: If equals one, this line contains the minimum order quantity and should be ordered prior to other lines of the same sku
Index

*Topic datasets
  unitsbro, 4

group_moq, 2

knapsack, 2

mknapsack, 3

moq_constraint, 4

unitsbro, 4