Package ‘critpath’

January 9, 2024

Type Package

Title Setting the Critical Path in Project Management

Version 0.2.2

Author Adam Kucharski

Maintainer Adam Kucharski <aqcharski@interia.pl>

Description

Solving the problem of project management using CPM (Critical Path Method), PERT (Program Evaluation and Review Technique) and LESS (Least Cost Estimating and Scheduling) methods. The package sets the critical path, schedule and Gantt chart. In addition, it allows to draw a graph even with marked critical activities. For more information about project management see: Taha H. A. ``Operations Research. An Introduction'' (2017, ISBN:978-1-292-16554-7), Rama Murthy P. ``Operations Research'' (2007, ISBN:978-81-224-2944-2), Yuval Cohen & Arik Sadeh (2006) ``A New Approach for Constructing and Generating AOA Networks'', Journal of Engineering, Computing and Architecture 1. 1-13, Konarzewska I., Jewczak M., Kucharski A. (2020, ISBN:978-83-8220-112-3), Miszczyńska D., Miszczyński M. ``Wybrane metody badań operacyjnych'' (2000, ISBN:83-907712-0-9).

License GPL-2

Encoding UTF-8

LazyData true

Imports DiagrammeR, ggplot2, reshape2, dplyr, stringr

Suggests knitr, rmarkdown

VignetteBuilder knitr

Depends R (>= 4.0.0)

Date 2024-01-06

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2024-01-09 20:20:05 UTC

R topics documented:

`cpmexample1` *Dataset for the CPM method*

Description

Fictitious data that is used in the examples. 6 activities, 5 nodes. In this dataset, the activities occur on the edges.

Usage

```
cpmexample1
```

Format

A data frame composed of predetermined columns:

from starting activity node

to final activity node

label activity label

time duration of the activity

`cpmexample2` *Dataset for the CPM method*

Description

Example from Miszczyńska D., Miszczyński M. "Wybrane metody badań operacyjnych" (2000, ISBN:83-907712-0-9). 10 activities, 8 nodes. In this dataset, the activities occur on the edges and a list of direct predecessors has been added.

Usage

```
cpmexample2
```

Format

A data frame composed of predetermined columns:

label activity label

pred preceding activities

time duration of the activity

`lessexample1` *Dataset for the LESS method*

Description

Fictitious data that is used in the examples. 6 activities, 5 nodes. In this dataset, the activities occur on the edges

Usage

```
lessexample1
```

Format

A data frame composed of predetermined columns:

from starting activity node

to final activity node

label activity label

time normal duration of the activity

bound_time the shortest duration of the activity

norm_cost normal cost of the activity

bound_cost boundary cost of the activity

lessexample2 *Dataset for the LESS method*

Description

Example from Miszczyńska D., Miszczyński M. "Wybrane metody badań operacyjnych" (2000, ISBN:83-907712-0-9). In this dataset, the activities occur on the edges and a list of direct predecessors has been added.

Usage

```
lessexample2
```

Format

A data frame composed of predetermined columns:

label activity label

pred preceding activities

time normal duration of the activity

bound_time the shortest duration of the activity

norm_cost normal cost of the activity

bound_cost boundary cost of the activity

pertexample1 *Dataset for the PERT method*

Description

Fictitious data that is used in the examples. 9 activities, 8 nodes. In this dataset, the activities occur on the edges

Usage

```
pertexample1
```

Format

A data frame composed of predetermined columns:

from starting activity node

to final activity node

label activity label

opt_time optimistic duration of activity

likely_time the most likely duration of the activity

pes_time pesimistic duration of activity

pertexample2 *Dataset for the PERT method*

Description

Example from Miszczyńska D., Miszczyński M. "Wybrane metody badań operacyjnych" (2000, ISBN:83-907712-0-9). 10 activities, 8 nodes. In this dataset, the activities occur on the edges and a list of direct predecessors has been added.

Usage

```
pertexample2
```

Format

A data frame composed of predetermined columns:

label activity label

pred preceding activities

opt_time optimistic duration of activity

likely_time the most likely duration of the activity

pes_time pesimistic duration of activity

PERT_newprob *Probability for the given directive term*

Description

Probability for the given directive term

Usage

```
PERT_newprob(new_DT, yourlist)
```

Arguments

`new_DT`	The given project completion date. The parameter must be greater than zero.
`yourlist`	List of objects that make up the solution to the project management problem.

Value

This function calculates the probability of completing the project within the time specified by the user. A normal distribution was assumed.

Examples

```
y <- solve_pathAOA(pertexample1, deterministic = FALSE)
PERT_newprob(new_DT = 30, y)
```

PERT_newtime *A new directive term for any probability*

Description

A new directive term for any probability

Usage

```
PERT_newtime(new_prob = 0.5, yourlist)
```

Arguments

new_prob	Probability of the project completion. Default set to 0.5.
yourlist	List of objects that make up the solution to the project management problem.

Value

This function computes a new directive term for a probability given by the user. A normal distribution was assumed.

Examples

```
y <- solve_pathAOA(pertexample1, deterministic = FALSE)
PERT_newtime(new_prob = 0.3, y)
```

plot_alap *An ALAP chart*

Description

An ALAP chart

Usage

```
plot_alap(yourlist, show_dummy = FALSE, bar_size = 10)
```

Arguments

yourlist	List of objects that make up the solution to the project management problem.
show_dummy	Decides whether dummy activities should be included in the chart. If so, set it to TRUE (set to FALSE by default).
bar_size	Thickness of the bar drawn for activity (set to 10 by default).

Value

Draws an ALAP (activities start and finish As Late As Possible) chart broken down into critical ("CR") and non-critical ("NC") activities. Marks total float.

Examples

```
x <- solve_pathAOA(cpmexample1, deterministic = TRUE)
plot_alap(x)
```

---

plot_asap *An ASAP chart*

Description

An ASAP chart

Usage

```
plot_asap(yourlist, show_dummy = FALSE, bar_size = 10)
```

Arguments

yourlist	List of objects that make up the solution to the project management problem.
show_dummy	Decides whether dummy activities should be included in the chart. If so, set it to TRUE (set to FALSE by default).
bar_size	Thickness of the bar drawn for activity (set to 10 by default).

Value

Draws an ASAP (activities start and finish As Soon As Possible) chart broken down into critical ("CR") and non-critical ("NC") activities. Marks total floats.

Examples

```
x <- solve_pathAOA(cpmexample1, deterministic = TRUE)
plot_asap(x)
```

plot_gantt *A Gantt chart*

Description

A Gantt chart

Usage

```
plot_gantt(yourlist, show_dummy = FALSE, bar_size = 10)
```

Arguments

`yourlist`	List of objects that make up the solution to the project management problem.
`show_dummy`	Decides whether dummy activities should be included in the chart. If so, set it to TRUE (set to FALSE by default).
`bar_size`	Thickness of the bar drawn for activity (set to 10 by default).

Value

Draws a Gantt chart broken down into critical ("CR") and non-critical ("NC") activities. Marks total floats.

Examples

```
x <- solve_pathAOA(cpmexample1, deterministic = TRUE)
plot_gantt(x)
```

plot_graphAOA *A graph of connections between nodes*

Description

A graph of connections between nodes

Usage

```
plot_graphAOA(input_data, predecessors = FALSE, solved = NULL, fixed_seed = 23)
```

Arguments

`input_data`	Data frame describing the problem.
`predecessors`	TRUE if the user data contains a list of immediately preceding activities
`solved`	List of objects that make up the solution to the project management problem.
`fixed_seed`	Optional parameter setting random seed to user value to get similar looking plots each time the function is run (set to 23 by default).

Value

The function draws a graph showing dependencies between nodes. The "solved" parameter determines whether there is a critical path in the graph. In that case, you must solve the problem first. In the examples below, the function first draws the graph only on the basis of the data frame and then after determining the critical path.

Examples

```
plot_graphAOA(cpmexample1)
x <- solve_pathAOA(cpmexample1, TRUE)
plot_graphAOA(solved = x)
```

plot_norm *The cumulative distribution function of the normal distribution*

Description

The cumulative distribution function of the normal distribution

Usage

```
plot_norm(yourlist)
```

Arguments

yourlist	List of objects making up the solution to the project management problem

Value

Draws a graph of the normal distribution with the expected directive term from the PERT method and the standard deviation for this term. The chart also includes lines indicating the schedules of the risk-taker and the belayer.

Examples

```
y <- solve_pathAOA(pertexample1, deterministic = FALSE)
plot_norm(y)
```

plot_TC *Total cost change plot*

Description

Total cost change plot

Usage

```
plot_TC(your_list)
```

Arguments

your_list	List containing solved problem

Value

Based on the results of the LESS method, a graph of the total cost value of all iterations is created

Examples

```
z <- solve_lessAOA(lessexample1, 50, 15)
plot_TC(z)
```

solve_lessAOA *Determines the solution using the LESS method. Relationships between activities can be given as a list of predecessors or start and end node numbers.*

Description

Determines the solution using the LESS method. Relationships between activities can be given as a list of predecessors or start and end node numbers.

Usage

```
solve_lessAOA(input_data, ICconst, ICslope, predecessors = FALSE)
```

Arguments

input_data Data frame containing the graph structure and activity durations. For the LESS method and start/end nodes you need 7 columns (the order matters):

1. `from` The number of the node where the activity starts.
2. `to` The number of the node where the activity ends.
3. `label` Activity labels.
4. `time` Normal duration of activities.

5. `crash_time` Crash (the shortest possible) duration of activities.
6. `norm_cost` Normal costs.
7. `crash_cost` Crash costs.

For the LESS method and predecessors list you need 6 columns (the order matters):

1. `label` Activity labels.
2. `pred` List of predecessors.
3. `time` Normal duration of activities.
4. `crash_time` Crash (the shortest possible) duration of activities.
5. `norm_cost` Normal costs.
6. `crash_cost` Crash costs.

`ICconst` Intercept of the indirect cost function.

`ICslope` Slope of the indirect cost function.

`predecessors` TRUE if the user data contains a list of immediately preceding activities If set to `FALSE` (default), start nad end nodes are used. If is set to `TRUE`, predecessors list is used.

Value

A list made of a graph and a result set.

Examples

```
z <-  solve_lessAOA(lessexample1, 50, 15)
```

`solve_pathAOA` *Finds a solution using CPM and PERT methods. Relationships between activities can be given as a list of predecessors or start and end node numbers.*

Description

Finds a solution using CPM and PERT methods. Relationships between activities can be given as a list of predecessors or start and end node numbers.

Usage

```
solve_pathAOA(
  input_data,
  deterministic = TRUE,
  predecessors = FALSE,
  pert_param = 0
)
```

Arguments

`input_data` Data frame containing the structure of the graph and the duration of the activity. For the CPM method and start/end nodes you need 4 columns (the order is important, not the name of the column):

1. `from` The number of the node where the activity starts.
2. `to` The number of the node where the activity ends.
3. `label` Activity labels.
4. `time` Activities durations.

For the CPM method and predecessors list you need 3 columns (the order is important, not the name of the column):

1. `label` Activity labels.
2. `pred` List of predecessors.
3. `time` Activities durations.

For the PERT method and start/end nodes you need 6 columns (the order is important, not the name of the column):

1. `from` The number of the node where the activity starts.
2. `to` The number of the node where the activity ends.
3. `label` Activity labels.
4. `opt_time` Optimistic duration of activities.
5. `likely_time` The most likely duration of the activity.
6. `pes_time` Pessimistic duration of activities.

For the PERT method and predecessors list you need 5 columns (the order is important, not the name of the column):

1. `label` Activity labels.
2. `pred` List of predecessors.
3. `opt_time` Optimistic duration of activities.
4. `likely_time` The most likely duration of the activity.
5. `pes_time` Pessimistic duration of activities.

`deterministic` A logical parameter specifying the solution method. If set to `TRUE` (default), the CPM method is used. If is set to `FALSE`, the PERT method is used.

`predecessors` TRUE if the user data contains a list of immediately preceding activities If set to `FALSE` (default), start nad end nodes are used. If is set to `TRUE`, predecessors list is used.

`pert_param` A parameter that controls the method of calculating the expected value and variance in the PERT method. 0 - classic formula (default), 1 - 1st and 99th percentile of the beta distribution, 2 - 5th and 95th percentile of the beta distribution, 3 - 5th and 95th percentiles of the beta distribution with modification by (Perry and Greig, 1975), 4 - Extended Pearson's and Tukey's formula (Pearson and Tukey, 1965), 5 - Golenko-Ginzburg's full formula (Golenko-Ginzburg, 1988), 6 - Golenko-Ginzburg's reduced formula (Golenko-Ginzburg, 1988), 7 - Farnum's and Stanton's formula (Farnum and Stanton, 1987).

Value

The list is made of a graph, schedule and selected partial results.

Examples

```
x <- solve_pathAOA(cpmexample1, deterministic = TRUE)
y <- solve_pathAOA(pertexample1, deterministic = FALSE)
x <- solve_pathAOA(cpmexample2, deterministic = TRUE, predecessors = TRUE)
y <- solve_pathAOA(pertexample2, deterministic = FALSE, predecessors = TRUE)
```

Index