

Package ‘ifm’

May 1, 2016

Type Package

Title Set of functions for financial evaluation of Software Projects

Version 1.0

Date 2016-04-20

Author Eber Schmitz

Maintainer Alexandre Costa <afcosta@br.ibm.com> and
Antoanne Pontes <antoanne@ufrj.br> and
Eduardo Chiote <eduardochiote@gmail.com>

Description R package with a set of functions for financial evaluation of
Software Project.

License LGPL (>= 2.1)

URL <https://github.com/afcosta-ibm/ifm>

BugReports <https://github.com/afcosta-ibm/ifm/issues>

NeedsCompilation no

RoxygenNote 5.0.1

#Imports igraph

R topics documented:

ifm-package	2
cpm.all.schedule	2
critical.path.method	3
discount.rate.vector	4
draw.cfs	5
future.value	6
inflation.free.interest.rate	6
net.future.value	7
net.present.value	8
present.value	8

Index	10
--------------	-----------

ifm-package

Set of functions for financial evaluation of Software Projects

Description

R package with a set of functions for financial evaluation of Software Project.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Eber Schmitz

Maintainer: Alexandre Costa <afcosta@br.ibm.com> and Antoanne Pontes <antoanne@ufrj.br>
and Eduardo Chiote <eduardochiote@gmail.com>

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~

Examples

examples here...

cpm.all.schedule

Generates all possible schedules for a cpm network

Description

Generates all possible schedules for a cpm network

Usage

```
cpm.all.schedule(est, slack)
```

Arguments

est	early start time vector
slack	activities slack

Value

Matrix with all minimum makespan cpm start time schedules

See Also

Other scheduling: [critical.path.method](#)

Examples

```
#Use critical.path.method function to calculate a set of project
#activities:

ex.cpm.activities.duration <- c(1,4,5,7,2,3,1)
ex.cpm.activities.successors <- list(c(2,3), 4, c(4,5), 6, 7, 7, c(0))
ex.cpm <- critical.path.method(ex.cpm.activities.duration,
                              ex.cpm.activities.successors)

# Now, that we have the CPM vector with:
# - est (Early Start Time) - ex.cpm[[1]]
# - eft (Early Finish Time) - ex.cpm[[2]]
# - lst (Late Start Time) - ex.cpm[[3]]
# - lft (Late Finish Time) ex.cpm[[4]]

ex.cpm.activities.schedule <- cpm.all.schedule(ex.cpm[[1]],
                                              ex.cpm[[3]] - ex.cpm[[1]])
```

`critical.path.method` *Scheduling a set of project activities*

Description

The Critical Path Method or Critical Path Analysis, is a mathematically based algorithm for scheduling a set of project activities.

CPM will get how long your complex project will take to complete and which activities are "critical," meaning that they have to be done on time or else the whole project will take longer.

CPM calculates:

As input, CPM will receive:

- A list of all activities required to complete the project;
- The time (duration) that each activity will take to completion;
- The dependencies between the activities.

Usage

```
critical.path.method(activities.duration, activities.successors)
```

Arguments

`activities.duration`

Vector with activities duration.

`activities.successors`

Vector with dependencies between activities.

Value

The optimized sequence of activities that must be performed to guarantee the shortest duration.

See Also

Other scheduling: [cpm.all.schedule](#)

Examples

```
ex.cpm.activities.duration <- c(1,4,5,7,2,3,1)
ex.cpm.activities.successors <- list(c(2,3), 4, c(4,5), 6, 7, 7, c(0))
ex.cpm <- critical.path.method(ex.cpm.activities.duration,
                             ex.cpm.activities.successors)
```

discount.rate.vector *Vectorize the Discount Rate*

Description

Generate a vector with discount rate to be applied to each of the time periods.

Usage

```
discount.rate.vector(interest.rate, number.of.periods,
  begin.of.period = FALSE)
```

Arguments

interest.rate A number that represents the nominal Interest Rate, presented by year.

number.of.periods Times that interest rate should be applied.

begin.of.period A boolean that represents if the Tax Rate will be applied at the beginning of period. FALSE by default.

See Also

Other financial: [draw.cfs](#), [future.value](#), [inflation.free.interest.rate](#), [net.future.value](#), [net.present.value](#), [present.value](#)

Examples

```
ex.disc.vector <- discount.rate.vector(6.19, 12)
```

draw.cfs	<i>Draw the graph of cash flow in order to facilitate the study and the effects of the analysis of a certain application.</i>
----------	---

Description

Cash flow is a mathematical concept that can be plotted in order to facilitate the study and the effects of the analysis of a certain application, which may be an investment loan, finance, etc.

Usage

```
draw.cfs(cfs, gt = "Cash Flow Graphic")
```

Arguments

cfs	A vector with a series of cash flows.
gt	A title for the graph.

Details

Normally a cash flow contains inputs and outputs of capital, marked in the timeline starting at $t = 0$.

A typical example is the graph that represents a bank loan held by a form of business she shall return this loan in n equal installments over the following months.

And we note that the value is entered in the company's cash (cash was positive) and S_1, S_2, \dots, S_n are the values of the parcels will leave the company's cash (negative).

The fact that each arrow is pointing upward (positive) or down (negative), it is assumed by convention, and the cash flow will depend on who receives or pays the Capital at a certain time, and:

$t = 0$ indicates the current day;

E_k is the capital input at a time k ;

S_k is the capital output at a time k .

See Also

Other financial: [discount.rate.vector](#), [future.value](#), [inflation.free.interest.rate](#), [net.future.value](#), [net.present.value](#), [present.value](#)

Examples

```
ex.cfs <- c(-2000,1000,1500,-500,500)
draw.cfs(ex.cfs, 'My Cash Flow')
```

future.value	<i>Calculate the future value of an asset at a specific date.</i>
--------------	---

Description

It measures the nominal future sum of money that a given sum of money is "worth" at a specified time in the future assuming a certain interest rate, or more generally, rate of return.

Usage

```
future.value(present.value, interest.rate, number.of.periods)
```

Arguments

`present.value` A number that represents the present value of the money.
`interest.rate` A number that represents the interest rate.
`number.of.periods` A number that represent the number of periods.

See Also

Other financial: [discount.rate.vector](#), [draw.cfs](#), [inflation.free.interest.rate](#), [net.future.value](#), [net.present.value](#), [present.value](#)

Examples

```
ex.fv <- future.value(1000, 1.1425, 12)
```

inflation.free.interest.rate	<i>Calculate the Inflation-free Interest Rate.</i>
------------------------------	--

Description

Calculate the Inflation-free Interest Rate.

Usage

```
inflation.free.interest.rate(interest.rate = 14.25, inflation.rate = 7.59)
```

Arguments

`interest.rate` A number that represents the nominal Interest Rate, presented by year.
`inflation.rate` A number that represents the Inflation Rate, presented by year.

See Also

Other financial: [discount.rate.vector](#), [draw.cfs](#), [future.value](#), [net.future.value](#), [net.present.value](#), [present.value](#)

Examples

```
ex.ifir <- inflation.free.interest.rate(14.25, 12)
```

net.future.value	<i>Vectorize the Future Value</i>
------------------	-----------------------------------

Description

Vectorize the Future Value

Usage

```
net.future.value(cfs, interest.rate, begin.of.period = FALSE)
```

Arguments

cfs	A vector with a series of cash flows.
interest.rate	A number that represents the nominal Interest Rate, presented by year.
begin.of.period	A boolean that represents if the Tax Rate will be applied at the beginning of period. FALSE by default.

See Also

Other financial: [discount.rate.vector](#), [draw.cfs](#), [future.value](#), [inflation.free.interest.rate](#), [net.present.value](#), [present.value](#)

Examples

```
ex.nfv <- net.future.value(c(-350,100,200,150,75), 6.19, TRUE)
```

net.present.value	<i>Vectorize the Present Value</i>
-------------------	------------------------------------

Description

Vectorize the Present Value

Usage

```
net.present.value(cfs, interest.rate, begin.of.period = FALSE)
```

Arguments

cfs	A vector with a series of cash flows.
interest.rate	A number that represents the nominal Interest Rate, presented by year.
begin.of.period	A boolean that represents if the Tax Rate will be applied at the beginning of period. FALSE by default.

See Also

Other financial: [discount.rate.vector](#), [draw.cfs](#), [future.value](#), [inflation.free.interest.rate](#), [net.future.value](#), [present.value](#)

Examples

```
ex.npv <- net.present.value(c(-350,100,200,150,75), 6.19, TRUE)
```

present.value	<i>Calculate the present value of an asset at a specific date.</i>
---------------	--

Description

In economics, present value, also known as present discounted value, is the value of an expected income stream determined as of the date of valuation. The present value is always less than or equal to the future value because money has interest-earning potential, a characteristic referred to as the time value of money, except during times of negative interest rates, when the present value will be less than the future value.

Usage

```
present.value(future.value, interest.rate, number.of.periods)
```

Arguments

future.value	A number that represents the future value of the money.
interest.rate	A number that represents the interest rate.
number.of.periods	A number that represent the number of periods.

See Also

Other financial: [discount.rate.vector](#), [draw.cfs](#), [future.value](#), [inflation.free.interest.rate](#), [net.future.value](#), [net.present.value](#)

Examples

```
ex.pv <- present.value(1000, 1.1425, 12)
```

Index

- *Topic **activities**
 - cpm.all.schedule, 2
 - critical.path.method, 3
- *Topic **critical**
 - cpm.all.schedule, 2
 - critical.path.method, 3
- *Topic **discount**
 - discount.rate.vector, 4
- *Topic **drawcfs**
 - draw.cfs, 5
- *Topic **futureValue**
 - future.value, 6
 - net.future.value, 7
- *Topic **inflation-free,**
 - inflation.free.interest.rate, 6
- *Topic **interest**
 - discount.rate.vector, 4
 - inflation.free.interest.rate, 6
- *Topic **package**
 - ifm-package, 2
- *Topic **path,**
 - cpm.all.schedule, 2
 - critical.path.method, 3
- *Topic **presentValue**
 - net.present.value, 8
 - present.value, 8
- *Topic **project**
 - cpm.all.schedule, 2
 - critical.path.method, 3
- *Topic **rate,**
 - discount.rate.vector, 4
- *Topic **rate**
 - discount.rate.vector, 4
 - inflation.free.interest.rate, 6
- *Topic **scheduling,**
 - cpm.all.schedule, 2
 - critical.path.method, 3

cpm(critical.path.method), 3
cpm.all.schedule, 2, 4
cpm_all_schedule(cpm.all.schedule), 2
critical.path.method, 3, 3
critical_path_method
 (critical.path.method), 3

disc(discount.rate.vector), 4
discount.rate.vector, 4, 5–9
draw.cfs, 4, 5, 6–9
draw_cfs(draw.cfs), 5
drawCfs(draw.cfs), 5

future.value, 4, 5, 6, 7–9
future_value(future.value), 6
futureValue(future.value), 6

genAllCpmSched(cpm.all.schedule), 2

IFIR(inflation.free.interest.rate), 6
IfIR(inflation.free.interest.rate), 6
ifir(inflation.free.interest.rate), 6
ifm(ifm-package), 2
ifm-package, 2
inflation.free.interest.rate, 4–6, 6, 7–9

net.future.value, 4–7, 7, 8, 9
net.present.value, 4–7, 8, 9
nfv(net.future.value), 7
npv(net.present.value), 8

present.value, 4–8, 8
present_value(present.value), 8
presentValue(present.value), 8