Package 'ifm'

April 26, 2016

Type Package

Title Set of functions for financial evaluation of Software Projects

Version 1.0

Date 2016-04-20

Author Eber Schmitz

Maintainer Antoanne Pontes <antoanne@ufrj.br>

Description R packeage 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 *Set of functions for financial evaluation of Software Projects*

Description

R packeage 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: Antoanne Pontes <antoanne@ufrj.br>

References

~~ Literature or other references for background information ~~

See Also

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

Examples

```
## examples here...
```

disc *~~function to do ... ~~*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
disc(r, n, bop = FALSE)
```

Arguments

r	~~Describe r here~~
n	~~Describe n here~~
bop	~~Describe bop here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1	Description of 'comp1'
comp2	Description of 'comp2'

Note

~~further notes~~

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as help, ~~~

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (r, n, bop = FALSE)
{
    t < -1/(1 + r)
    e <- if (bop) {
        0:(n - 1)
    }
    else {
        1:n
    }
    return(t^e)
  }
```

`draw.cfs` *Draw the graph of cash flow.*

Description

Draw the graph of cash flow.

Usage

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

Arguments

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

See Also

Other financial: `future.value`, `inflation.free.interest.rate`

Examples

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

`future.value` *Calculate the future value of an asset at a specific date. 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.*

Description

Calculate the future value of an asset at a specific date. 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: draw.cfs, inflation.free.interest.rate

Examples

```
fv.1 <- future.value(1000, 1.1425, 12)
print(fv.1)
```

`inflation.free.interest.rate` *Calculate the Inflation-free Interest Rate.*

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: draw.cfs, future.value

Examples

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

`nfv` *~~function to do ... ~~*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
nfv(cfs, r, bop = TRUE)
```

Arguments

cfs	~~Describe cfs here~~
r	~~Describe r here~~
bop	~~Describe bop here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1	Description of 'comp1'
comp2	Description of 'comp2'

Note

~~further notes~~

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as help, ~~~

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (cfs, r, bop = TRUE)
{
    e <- if (bop) {
        0:(length(cfs) - 1)
    }
    else {
        1:(length(cfs))
    }
    tax <- (1 + (r/100))^e
    return(cfs * tax)
  }
```

npv *~~function to do ... ~~*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
npv(cfs, r, bop = TRUE)
```

Arguments

cfs	~~Describe cfs here~~
r	~~Describe r here~~
bop	~~Describe bop here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1	Description of 'comp1'
comp2	Description of 'comp2'

Note

~~further notes~~

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as help, ~~~

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (cfs, r, bop = TRUE)
{
    e <- if (bop) {
        0:(length(cfs) - 1)
    }
    else {
        1:(length(cfs))
    }
    tax <- (1/((1 + (r/100))^e))
    return(cfs * tax)
  }
```

---

pv *pv.*

Description

pv.

Usage

```
pv(Fv, r, n)
```

Index