Package 'imf'

April 20, 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/antoanne/ifm

BugReports https://github.com/antoanne/ifm/issues

NeedsCompilation no

R topics documented:

`imf-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)
  }
```

`drawcfs` *~~function to do ... ~~*

Description

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

Usage

```
drawcfs(cfs)
```

Arguments

`cfs`	~~Describe `cfs` 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)
{
    plot(cfs, xlab = "Periodo", ylab = "y")
    abline(h = c(0))
    title("Fluxo de Caixa")
    count <- 0
    for (i in cfs) {
        count <- count + 1
        if (i != 0) {
            arrows(count, 0, count, i, length = 0.1, angle = 20)
        }
    }
  }
```

---

fv *~~function to do ... ~~*

Description

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

Usage

```
fv(Pv, r, n)
```

Arguments

Pv	~~Describe Pv here~~
r	~~Describe r here~~
n	~~Describe n 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 (Pv, r, n)
{
    return(Pv * (1 + (r/100))^n)
  }
```

`ifir` *~~ Function to calculate inflation free interest rate. ~~*

Description

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

Usage

```
ifir(r, f)
```

Arguments

`r`	~~Describe `r` here~~
`f`	~~Describe `f` 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, f)
{
    return((r - f)/(1 + f))
  }
```

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

Description

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

Usage

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

Arguments

Fv	~~Describe Fv here~~
r	~~Describe r here~~
n	~~Describe n 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 (Fv, r, n)
{
    return(Fv/(1 + (r/100))^n)
  }
```

Index