

# Package ‘financial’

February 14, 2012

**Version** 0.1

**Date** 2006-04-09

**Title** Solving financial problems in R

**Author** Lukasz Komsta <luke@novum.am.lublin.pl>

**Maintainer** Lukasz Komsta <luke@novum.am.lublin.pl>

**Depends** R (>= 2.0)

**Description** Time value of money, cash flows and other financial functions.

**License** GPL (>= 2)

**URL** <http://www.r-project.org>, <http://www.komsta.net/>

**Repository** CRAN

**Date/Publication** 2006-04-11 15:23:40

## R topics documented:

cf	2
ireff	3
plot.cf	4
plot.tvm	5
sppv	5
summary.cf	6
summary.tvm	7
tvm	8
update.cf	9
update.tvm	10
<b>Index</b>	<b>11</b>

cf

*Cash Flow model***Description**

Calculates a Cash Flow model from a vector of flows.

**Usage**

```
cf(x, i = NULL, safe = NULL, rein = safe)
```

**Arguments**

x	a vector of cash flows.
i	a vector of periodic rates used in calculating NPV, NFV and NUS table.
safe	a vector of safe periodic rates using in calculating MIRR.
rein	a vector of reinvestment periodic rates using in calculating MIRR.

**Details**

The negative values are money paid out, the positive values are money received in.

IRR (Internal rate of return) is calculated by solving polynomial, using [polyroot](#) function. If IRR equation have several real solutions, all are found.

NPV extremes are found in the same method, after obtaining first-order derivative of IRR equation.

MIRR table and NPV table are created only if i, safe and rein are given (are not null).

**Value**

An object of class "cf", being a list of the following compounds:

cf	a vector of cash flows.
mirr	a matrix containing for each safe and reinvestment rate pair - modified internal rate of return values. If not calculated, has NULL value.
tab	a matrix containing for each periodic rate (i) - NPV, NFV and NUS value. If not calculated, set to NULL.
irr	a vector of IRR values - real rates giving NPV equal to zero. The number of values are dependent of sign changes in cash flow series (one or more real roots of polynomial).
ext	a vector of rate values giving extremes in NPV - maxima or minima. May have no values, if they are no sign changes in cash flows.

**Author(s)**

Lukasz Komsta

**See Also**[summary.cf](#)**Examples**

```
y = cf(c(-2500,6250,-3800),1:10,2:11,4:13)
y
summary(y)
plot(y)
plot(y,type="n")
```

---

**ireff***Effective and nominal financial rates*

---

**Description**

Calculates effective and nominal rates for periodic and continuous compounding.

**Usage**

```
ireff(nom, p)
irnom(eff, p)
```

**Arguments**

nom	a vector of nominal rates.
eff	a vector of effective rates.
p	a vector of times of compounding, may be Inf for continuous compounding.

**Value**

A vector of nominal (irnom) or effective (ireff) rates.

**Author(s)**

Lukasz Komsta

**See Also**[irnom](#)**Examples**

```
ireff(20,c(1:20,Inf))
irnom(20,c(1:20,Inf))
```

---

`plot.cf`*Plot of cash flow model*

---

**Description**

Plots bar plot of the cash flows or NPV versus periodic rate.

**Usage**

```
plot.cf(x, type = c("bar", "npv"), ...)
```

**Arguments**

<code>x</code>	an object of class "cf", obtained using <a href="#">cf</a> .
<code>type</code>	type of the plot.
<code>...</code>	additional arguments passed to <a href="#">barplot</a> internally.

**Details**

The "bar" plot displays all cash flows as bars, their cumulative sum as a line-plot and NPV as horizontal lines.

The "npv" plot displays dependence between rate and NPV in range 0 - 100 percent. The roots (IRR values) and extremes are indicated by vertical lines.

**Author(s)**

Lukasz Komsta

**See Also**

[cf](#)

**Examples**

```
y = cf(c(-2500, 6250, -3800), 1:10, 2:11, 4:13)
plot(y)
plot(y, type="npv")
```

---

`plot.tvm`*Plot TVM models*

---

**Description**

Makes bar plots of TVM models.

**Usage**

```
plot.tvm(x, row = 1, ...)
```

**Arguments**

<code>x</code>	an object of class "tvm" containing time value of money model.
<code>row</code>	a row number in the model to plot.
<code>...</code>	additional arguments passed to <code>barplot</code> internally.

**Details**

The present value, payments and future value are plotted as bars. Cumulative sum of these are indicated by a line.

**Author(s)**

Lukasz Komsta

**See Also**

[tvm](#)

**Examples**

```
y=tvm(pv=10000, i=10, n=10, pmt=NA)
plot(y)
```

---

`sppv`*Single Payment or Uniform Series Present and Future Values*

---

**Description**

Calculate SPPV, SPFV, USPV and USFV factors from given rates and numbers of periods.

**Usage**

```
sppv(i, n)
spfv(i, n)
uspv(i, n)
usfv(i, n)
```

**Arguments**

i	a vector of rates (in percents).
n	a vector of periods.

**Value**

A vector of SPPV, SPFV, USPV or USFV values.

**Author(s)**

Lukasz Komsta

**Examples**

```
spfv(10, 1:100)
sppv(10, 1:100)
uspv(10, 1:100)
usfv(10, 1:100)
```

---

summary.cf

*Summarizing cash flow models*

---

**Description**

This function gives summary for each step in cash flow model.

**Usage**

```
summary.cf(object, flows = 2:length(object$cf), ...)
```

**Arguments**

object	an object of "cf" class, created using <code>cf</code> .
flows	a vector of first flow numbers to summarize, default is from first two flows to all flows.
...	additional arguments, currently ignored.

**Details**

Summarizing is done by printing all information for first "flows" flows. By default, the summary of first 2, 3, 4 ... n flows are printed.

**Author(s)**

Lukasz Komsta

**See Also**

[cf](#)

**Examples**

```
y = cf(c(-10,5,5,5,5,5))
summary(y)
```

---

summary.tvm

*Summarizing TVM model by printing its amortization table*

---

**Description**

Prints amortization table of TVM model.

**Usage**

```
summary.tvm(object, row = 1, ...)
```

**Arguments**

object	an object of class "tvm", created by <a href="#">tvm</a>
row	a row number to make summary, default first row.
...	additional arguments, currently ignored.

**Details**

A table containing balance, interest, principal and payment for each step is produced and printed.

**Author(s)**

Lukasz Komsta

**See Also**

[tvm](#)

**Examples**

```
y=tvm(pv=10000,i=10,n=10,pmt=NA)
summary(y)
```

tvm

*Fitting time value of money models***Description**

This function fits generalized time value of money models.

**Usage**

```
tvm(i = 0, n = 1, pv = 0, fv = 0, pmt = 0, days = 360/pyr, adv = 0, pyr = 12, cyr = pyr)
```

**Arguments**

<code>i</code>	a vector of nominal rates, as percentages.
<code>n</code>	a vector of period numbers.
<code>pv</code>	a vector of present values.
<code>fv</code>	a vector of future values.
<code>pmt</code>	a vector of payments.
<code>days</code>	a vector of days from begin of period to make payment. Default value is equivalent to END mode, 0 means BEGIN mode.
<code>adv</code>	a vector of numbers of payments made in advance.
<code>pyr</code>	a vector of numbers of payments per year.
<code>cyr</code>	a vector of numbers of compounding periods per year.

**Details**

In each variant (row) of arguments, one and EXACTLY one value must be set to NA, and this value is calculated from the others to "fit the model". For example, `pv` can be set to `c(10000, NA)` and `pmt` to `c(NA, -100)`. In first row `pmt` will be calculated, in the second - present value.

**Value**

A matrix of class "tvm" containing all the arguments (one row for each argument vector element).

**Author(s)**

Lukasz Komsta

**Examples**

```
y=tvm(pv=10000, i=1:10, n=10, pmt=NA)
y
update(y, pmt=-1000, pv=NA)
update(y, pmt=-1000, n=NA)
```

---

`update.cf`*Update a cash flow model*

---

**Description**

This function changes some data in cash flow model and recalculates it.

**Usage**

```
update.cf(object, flows = NULL, i = NULL, safe = NULL, rein = NULL, ...)
```

**Arguments**

<code>object</code>	an object of "cf" class, created by <code>cf</code> function.
<code>flows</code>	new updated flows.
<code>i</code>	a vector of new updated periodic rates.
<code>safe</code>	a vector of new updated safe MIRR rates.
<code>rein</code>	a vector of new updated reinvestment MIRR rates.
<code>...</code>	additional arguments, currently ignored.

**Details**

The function takes its first arguments, changing its given parameters, and returns recalculated model.

**Value**

An object of class "cf", see `link{cf}`.

**Author(s)**

Lukasz Komsta

**See Also**

[cf](#)

**Examples**

```
y = cf(c(-2500,6250,-3800),1:10,2:11,4:13)
y
update(y,i=11:20)
```

---

 update.tvm

*Update time value of money model*


---

**Description**

Changes something declared in TVM model and recalculates NA values.

**Usage**

```
update.tvm(object, i = NULL, n = NULL, pv = NULL, fv = NULL, pmt = NULL, days = NULL, adv = NULL, pyr = NULL, ...)
```

**Arguments**

object	an object of "tvm" class used to update.
i	a vector of nominal rates, as percentages.
n	a vector of period numbers.
pv	a vector of present values.
fv	a vector of future values.
pmt	a vector of payments.
days	a vector of days from begin of period to make payment. Default value is equivalent to END mode, 0 means BEGIN mode.
adv	a vector of numbers of payments made in advance.
pyr	a vector of numbers of payments per year.
cyr	a vector of numbers of compounding periods per year.
...	additional arguments, currently ignored.

**Details**

NA values must be REDECLARED in updating process, because TVM object cannot remember what parameters were NA!

**Value**

An updated object of "tvm" class, see [tvm](#)

**Author(s)**

Lukasz Komsta

**See Also**

[tvm](#)

**Examples**

```
y=tvm(pv=10000, i=1:10, n=10, pmt=NA)
update(y, pmt=-1000, n=NA)
```

# Index

## \*Topic **models**

- cf, [2](#)
- ireff, [3](#)
- plot.cf, [4](#)
- plot.tvm, [5](#)
- sppv, [5](#)
- summary.cf, [6](#)
- summary.tvm, [7](#)
- tvm, [8](#)
- update.cf, [9](#)
- update.tvm, [10](#)

barplot, [4](#), [5](#)

cf, [2](#), [4](#), [6](#), [7](#), [9](#)

ireff, [3](#)  
irnom, [3](#)  
irnom(ireff), [3](#)

plot.cf, [4](#)  
plot.tvm, [5](#)  
polyroot, [2](#)  
print.cf(cf), [2](#)  
print.tvm(tvm), [8](#)

spfv(sppv), [5](#)  
sppv, [5](#)  
summary.cf, [3](#), [6](#)  
summary.tvm, [7](#)

tvm, [5](#), [7](#), [8](#), [10](#)

update.cf, [9](#)  
update.tvm, [10](#)  
usfv(sppv), [5](#)  
uspv(sppv), [5](#)