

# Package ‘tvm’

May 8, 2026

**Type** Package

**Title** Time Value of Money Functions

**Version** 0.5.2

**Author** Juan Manuel Truppia

**Maintainer** Juan Manuel Truppia <jmtruppia@gmail.com>

**Description** Functions for managing cashflows and interest rate curves.

**License** MIT + file LICENSE

**Depends** R (>= 3.1.0)

**Suggests** testthat, knitr, markdown, rmarkdown

**Imports** ggplot2, reshape2, scales, stats, utils

**VignetteBuilder** knitr

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**URL** <https://bitbucket.org/juancentro/tvm>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-08-30 13:50:02 UTC

## Contents

adjust_disc . . . . .	2
cashflow . . . . .	2
cft . . . . .	3
disc_cf . . . . .	3
disc_value . . . . .	4
find_rate . . . . .	4
irr . . . . .	5
loan . . . . .	6
npv . . . . .	6
plot.rate_curve . . . . .	7

pmt . . . . .	8
rate . . . . .	8
rate_curve . . . . .	9
rem . . . . .	10
xirr . . . . .	11
xnpv . . . . .	11
[.rate_curve . . . . .	12

## Index 13

adjust\_disc *Adjusts the discount factors by a spread*

### Description

Adjusts the discount factors by a spread

### Usage

```
adjust_disc(fd, spread)
```

### Arguments

fd	vector of discount factors used to discount cashflows in 1:length(fd) periods
spread	effective spread

### Examples

```
adjust_disc(fd = c(0.99, 0.98), spread = 0.01)
```

cashflow *Get the cashflow for a loan*

### Description

Returns the cashflow for the loan, excluding the initial inflow for the loan taker

### Usage

```
cashflow(l)
```

### Arguments

l	The loan
---	----------

### Examples

```
l <- loan(rate = 0.05, maturity = 10, amt = 100, type = "bullet")
cashflow(l)
```

---

cft	<i>Calculates the Total Financial Cost (CFT)</i>
-----	--

---

**Description**

This is the IRR of the loan's cashflow, after adding all the extra costs

**Usage**

```
cft(amt, maturity, rate, up_fee = 0, per_fee = 0)
```

**Arguments**

amt	The amount of the loan
maturity	The maturity of the loan
rate	The loan rate, in effective rate
up_fee	The fee that the loan taker pays upfront
per_fee	The fee that the loan payer pays every period

**Details**

It is assumed that the loan has monthly payments The CFT is returned as an effective rate of periodicity equal to that of the maturity and the rate The interest is calculated over amt + fee

**Examples**

```
cft(amt = 100, maturity = 10, rate = 0.05, up_fee = 1, per_fee = 0.1)
```

---

disc_cf	<i>Value of a discounted cashflow</i>
---------	---------------------------------------

---

**Description**

Value of a discounted cashflow

**Usage**

```
disc_cf(fd, cf)
```

**Arguments**

fd	The discount factor vector
cf	The cashflow

**Examples**

```
disc_cf(fd = c(1, 0.99, 0.98, 0.97), cf = c(1, -0.3, -0.4, -0.6))
```

---

disc\_value                      *Calculates the present value of a cashflow*

---

### Description

Calculates the present value of a cashflow

### Usage

```
disc_value(r, cf, d = 1:length(cf))
```

### Arguments

r	A rate curve
cf	The vector of values corresponding to the cashflow
d	The periods on which the cashflow occurs. If missing, it is assumed that cf[i] occurs on period i

### Value

The present value of the cashflow

### Examples

```
r <- rate_curve(rates = c(0.1, 0.2, 0.3), rate_type = "zero_eff")
disc_value(r, cf = c(-1, 1.10), d = c(0,1))
disc_value(r, cf = c(-1, 1.15*1.15), d = c(0,2))
```

---

find\_rate                      *Find the rate for a loan given the discount factors*

---

### Description

Thru a root finding process, this function finds the rate that corresponds to a given set of discount factors, as for the loan to have the same present value discounted with the discount factors or with that constant rate

### Usage

```
find_rate(m, d, loan_type, interval = c(1e-06, 2), tol = 1e-08)
```

**Arguments**

m	The maturity of the loan
d	The discount factor vector
loan_type	One of the loan types
interval	The interval for the root finding process
tol	The tolerance for the root finding process

**Examples**

```
find_rate(m = 3, d = c(0.99, 0.98, 0.97), loan_type = "bullet")
```

---

irr	<i>The IRR is returned as an effective rate with periodicity equal to that of the cashflow</i>
-----	--

---

**Description**

Internal Rate of Return of a periodic cashflow (IRR)

**Usage**

```
irr(cf, ts = seq(from = 0, by = 1, along.with = cf), interval = c(-1, 10), ...)
```

**Arguments**

cf	The cashflow
ts	The times on which the cashflow occurs. It is assumed that cf[idx] happens at moment ts[idx]
interval	A length 2 vector that indicates the root finding algorithm where to search for the irr
...	Other arguments to be passed on to uniroot

**Examples**

```
irr(cf = c(-1, 0.5, 0.9), ts = c(0, 1, 3))
```

---

loan	<i>Creates an instance of a loan class</i>
------	--

---

**Description**

Creates an instance of a loan class

**Usage**

```
loan(rate, maturity, amt, type, grace_int = 0, grace_amort = grace_int)
```

**Arguments**

rate	The periodic effective rate of the loan
maturity	The maturity of the loan, measured in the same units as the periodicity of the rate
amt	The amount loaned
type	The type of loan. Available types are c("bullet", "french", "german")
grace_int	The number of periods that the loan doesn't pay interest and capitalizes it. Leave in 0 for zero loans
grace_amort	The number of periods that the loan doesn't amortize

**Examples**

```
loan(rate = 0.05, maturity = 10, amt = 100, type = "bullet")
```

---

npv	<i>Net Present Value of a periodic cashflow (NPV)</i>
-----	---

---

**Description**

Net Present Value of a periodic cashflow (NPV)

**Usage**

```
npv(i, cf, ts = seq(from = 0, by = 1, along.with = cf))
```

**Arguments**

i	The rate used to discount the cashflow. It must be effective and with a periodicity that matches that of the cashflow
cf	The cashflow
ts	The times on which the cashflow occurs. It is assumed that cf[idx] happens at moment ts[idx]. If empty, assumes that cf[idx] happens at period idx - 1

**Value**

The net present value at

**Examples**

```
npv(i = 0.01, cf = c(-1, 0.5, 0.9), ts = c(0, 1, 3))
```

---

plot.rate_curve	<i>Plots a rate curve</i>
-----------------	---------------------------

---

**Description**

Plots a rate curve

**Usage**

```
## S3 method for class 'rate_curve'
plot(x, rate_type = NULL, y_labs_perc = TRUE, y_labs_acc = NULL, ...)
```

**Arguments**

x	The rate curve
rate_type	The rate types to plot, in c("french", "fut", "german", "zero_eff", "zero_nom", "swap", "zero_cont")
y_labs_perc	If TRUE, the y axe is labeled with percentages
y_labs_acc	If y_labs_perc is TRUE, the accuracy for the percentages (i.e., 1 for xx%, 0.1 for xx.x%, 0.01 for xx.xx%, etc)
...	Other arguments (unused)

**Examples**

```
r <- rate_curve(rates = c(0.1, 0.2, 0.3), rate_type = "zero_eff")
plot(r)
## Not run:
plot(r, rate_type = "german")
plot(r, rate_type = c("french", "german"))

## End(Not run)
```

---

pmt	<i>The value of the payment of a loan with constant payments (french type amortization)</i>
-----	---

---

**Description**

The value of the payment of a loan with constant payments (french type amortization)

**Usage**

```
pmt(amt, maturity, rate)
```

**Arguments**

amt	The amount of the loan
maturity	The maturity of the loan
rate	The rate of the loan

**Details**

The periodicity of the maturity and the rate must match, and this will be the periodicity of the payments

**Examples**

```
pmt(amt = 100, maturity = 10, rate = 0.05)
```

---

rate	<i>The rate of a loan with constant payments (french type amortization)</i>
------	---

---

**Description**

The rate of a loan with constant payments (french type amortization)

**Usage**

```
rate(amt, maturity, pmt, extrema = c(1e-04, 1e+09), tol = 1e-04)
```

**Arguments**

amt	The amount of the loan
maturity	The maturity of the loan
pmt	The payments of the loan
extrema	Vector of length 2 that has the minimum and maximum value to search for the rate
tol	The tolerance to use in the root finding algorithm

**Details**

The periodicity of the maturity and the payment must match, and this will be the periodicity of the rate (which is returned as an effective rate)

**Examples**

```
rate(amt = 100, maturity = 10, pmt = 15)
```

---

rate_curve	<i>Creates a rate curve instance</i>
------------	--------------------------------------

---

**Description**

Creates a rate curve instance

**Usage**

```
rate_curve(
  rates = NULL,
  rate_type = "zero_eff",
  pers = 1:length(rates),
  rate_scale = 1,
  fun_d = NULL,
  fun_r = NULL,
  knots = seq.int(from = 1, to = max(pers), by = 1),
  functor = function(x, y) splinefun(x = x, y = y, method = "monoH.FC")
)
```

**Arguments**

rates	A rate vector
rate_type	The rate type. Must be one of c("fut", "zero_nom", "zero_eff", "swap", "zero_cont")
pers	The periods the rates correspond to
rate_scale	In how many periods is the rate expressed. For example, when measuring periods in days, and using annual rates, you should use 365. When measuring periods in months, and using annual rates, you should use 12. If no scaling, use 1.
fun_d	A discount factor function. fun_d(x) returns the discount factor for time x, vectorized on x
fun_r	A rate function. fun_r(x) returns the EPR for time x, vectorized on x
knots	The nodes used to bootstrap the rates. This is a mandatory argument if a rate function or discount function is provided
functor	A function with parameters x and y, that returns a function used to interpolate

**Note**

Currently a rate curve can only be built from one of the following sources

1. A discount factor function
2. A rate function and a rate type from the following types: "fut", "zero\_nom", "zero\_eff", "swap" or "zero\_cont"
3. A rate vector, a pers vector and a rate type as before

**Examples**

```
rate_curve(rates = c(0.1, 0.2, 0.3), rate_type = "zero_eff")
rate_curve(fun_r = function(x) rep_len(0.1, length(x)), rate_type = "swap", knots = 1:12)
rate_curve(fun_d = function(x) 1 / (1 + x), knots = 1:12)
```

---

 rem

---

*Remaining capital in a loan*


---

**Description**

The amount that has to be repayed at each moment in a loan, at the end of the period

**Usage**

```
rem(cf, amt, r)
```

**Arguments**

cf	The cashflow of the loan, not including the initial inflow for the loan taker
amt	The original amount of the loan
r	The periodic rate of the loan

**Examples**

```
rem(cf = rep_len(0.4, 4), amt = 1, r = 0.2)
```

---

xirr	<i>The IRR is returned as an effective annual rate</i>
------	--

---

**Description**

Internal Rate of Return of an irregular cashflow (IRR)

**Usage**

```
xirr(cf, d, tau = NULL, comp_freq = 1, interval = c(-0.99999, 10), ...)
```

**Arguments**

cf	The cashflow
d	The dates when each cashflow occurs. Same length as the cashflow. Only used if tau is NULL. Assumes act/365 fractions
tau	The year fractions when each cashflow occurs. Same length as the cashflow
comp_freq	The compounding frequency used. Most relevant cases are 1 for yearly, 2 twice a year, 4 quarterly, 12 monthly, 0 no compounding, Inf continuous
interval	A length 2 vector that indicates the root finding algorithm where to search for the irr
...	Other arguments to be passed on to uniroot

**Examples**

```
xirr(cf = c(-1, 1.5), d = Sys.Date() + c(0, 365))
```

---

xnpv	<i>Net Present Value of an irregular cashflow (NPV)</i>
------	---

---

**Description**

Net Present Value of an irregular cashflow (NPV)

**Usage**

```
xnpv(i, cf, d, tau = NULL, comp_freq = 1)
```

**Arguments**

i	The rate used to discount the cashflow
cf	The cashflow
d	The dates when each cashflow occurs. Same length as the cashflow. Only used if tau is NULL. Assumes act/365 fractions
tau	The year fractions when each cashflow occurs. Same length as the cashflow
comp_freq	The compounding frequency used. Most relevant cases are 1 for yearly, 2 twice a year, 4 quarterly, 12 monthly, 0 no compounding, Inf continuous

**Examples**

```
xnpv(i = 0.01, cf = c(-1, 0.5, 0.9), d = as.Date(c("2015-01-01", "2015-02-15", "2015-04-10")))
```

---

[.rate_curve	<i>Returns a particular rate or rates from a curve</i>
--------------	--

---

**Description**

Returns a particular rate or rates from a curve

**Usage**

```
## S3 method for class 'rate_curve'
r[rate_type = "zero_eff", x = NULL]
```

**Arguments**

r	The rate_curve object
rate_type	The rate type
x	The points in time to return

**Value**

If x is NULL, then returns a rate function of rate\_type type. Else, it returns the rates of rate\_type type and corresponding to time x

**Examples**

```
r <- rate_curve(rates = c(0.1, 0.2, 0.3), rate_type = "zero_eff")
r["zero_eff"]
r["swap",c(1.5, 2)]
```

# Index

[.rate\_curve, 12

adjust\_disc, 2

cashflow, 2

cft, 3

disc\_cf, 3

disc\_value, 4

find\_rate, 4

irr, 5

loan, 6

npv, 6

plot.rate\_curve, 7

pmt, 8

rate, 8

rate\_curve, 9

rem, 10

xirr, 11

xnpv, 11