Package 'treasuryTR'

May 28, 2021

Type Package

Title Generate Treasury Total Returns from Yield Data

Version 0.1.4

Maintainer Martin Geissmann <mg@econovo.ch>

Description Generate Total Returns (TR) from bond yield data with fixed maturity, e.g. reported treasury yields. The generated TR series are very close to alternative series that can be purchased (e.g. CRSP, Bloomberg), suggesting they are a high-quality alternative for those, see Swinkels (2019) <doi:10.3390/data4030091>.

URL https://github.com/mgei/treasuryTR

Depends R (>= 3.2.0), quantmod, zoo, dplyr

Imports xts(>= 0.9-0), lubridate

Suggests PerformanceAnalytics, tidyr, ggplot2, dataseries, knitr, rmarkdown

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Martin Geissmann [aut, cre]

Repository CRAN

Date/Publication 2021-05-28 08:50:02 UTC

R topics documented:

convexity 2
get_yields 3
mod_duration 4
tibble_to_xts 5
total_return 5
xts_to_tibble 7

Index **8**

`convexity` *Calculate the convexity of a bond*

Description

Calculates the convexity of a bond.

This function is normally used in combination with total_return() to compute bond total returns.

Usage

```
convexity(yields, maturity, format_out = "xts")
```

Arguments

`yields`	a series of yields
`maturity`	constant bond maturity in years
`format_out`	xts or tibble

Details

This function is normally used in combination with total_return() to compute bond total returns.

The convexity is the interest rate sensitivity of the modified duration.

$$convexity = C_1 - C_2$$

where

$$C_1 = \frac{2}{y_t^2}(1 - {z_t}^{-2M})$$

$$C_2 = \frac{2M}{y_t}{z_t}^{-2M-1}$$

$$z_t = 1 + \frac{y_t}{2}$$

M is the maturity in years (e.g. 10), y_t is the yield at time t.

Value

The convexity or a series of convexities of a bond with the given yield and maturity

`get_yields` *Load data from FRED*

Description

Load data from FRED

Usage

```
get_yields(
  series = "DGS10",
  na_locf = TRUE,
  percent_adjust = TRUE,
  format_out = "xts",
  ...
)
```

Arguments

`series`	The series code as found on https://fred.stlouisfed.org/, see details
`na_locf`	replace NA's with last observation
`percent_adjust`	divide raw data by 100
`format_out`	xts or tibble
`...`	Additional parameters handed to quantmod::getSymbols

Details

The function is a wrapper for quantmod::getSymbols().

Commonly used constant-maturity yield series are:

- DGS1MO: 1-Month Treasury Constant Maturity Rate
- DGS3MO: 3-Month Treasury Constant Maturity Rate
- DGS6MO: 6-Month Treasury Constant Maturity Rate
- DGS1: 1-Year Treasury Constant Maturity Rate
- DGS2: 2-Year Treasury Constant Maturity Rate
- DGS3: 3-Year Treasury Constant Maturity Rate
- DGS5: 5-Year Treasury Constant Maturity Rate
- DGS7: 7-Year Treasury Constant Maturity Rate
- DGS10: 10-Year Treasury Constant Maturity Rate
- DGS20: 20-Year Treasury Constant Maturity Rate
- DGS30: 30-Year Treasury Constant Maturity Rate

Value

The yields data as an xts object

Examples

```
# US 1-Month yields
yields_us_1m <- get_yields(series = "DGS1MO")
```

`mod_duration` *Calculate the modified duration of a bond*

Description

Calculate the modified duration of a bond

Usage

```
mod_duration(yields, maturity, format_out = "xts")
```

Arguments

`yields`	a series of yields
`maturity`	constant bond maturity in years
`format_out`	xts or tibble

Details

This function is normally used in combination with total_return() to compute bond total returns.

The modified duration is the interest rate sensitivity of the price of bond.

$$duration = \frac{1}{y_t} {z_t}^{2M}$$

with

$$z_t = 1 + \frac{y_t}{2}$$

M is the maturity in years (e.g. 10), y_t is the yield at time t.

Value

A series of modified duration

tibble_to_xts *Convert a tibble data frame to an xts object*

Description

Convert a tibble data frame to an xts object

Usage

```
tibble_to_xts(x)
```

Arguments

x	a tibble with a column 'date'

Value

An xts object

total_return *Calculate bond total returns from constant-maturity yield data*

Description

Calculate bond total returns from constant-maturity yield data

Usage

```
total_return(
  yields,
  maturity,
  mdur = mod_duration(yields, maturity),
  convex = convexity(yields, maturity),
  scale = 261,
  format_out = "xts"
)
```

Arguments

yields	xts series or vector or yields
maturity	bond constant-maturity in years
mdur	modified duration, by default calculated using mod_duration()
convex	convexity, by default calculated using convexity()
scale	number of periods in a year (for US treasury data daily scale = 261, weekly scale = 52, monthly scale = 12, quarterly scale = 4)
format_out	xts or tibble

Details

Calculate bond total returns from constant-maturity yield data.

$$R_t = yieldincome - duration \cdot \Delta y + \frac{1}{2} \cdot convexity \cdot (\Delta y)^2$$

where

$$yieldincome = (1 + y_t)^{\Delta t} - 1 \approx y_t \Delta t$$

$$duration = \frac{1}{y_t} {z_t}^{2M}$$

$$convexity = C_1 - C_2$$

and

$$C_1 = \frac{2}{y_t^2}(1 - {z_t}^{-2M})$$

$$C_2 = \frac{2M}{y_t} {z_t}^{-2M-1}$$

$$z_t = 1 + \frac{y_t}{2}$$

M is the maturity in years (e.g. 10), y_t is the yield at time t

Value

Bond total returns

Examples

```
library(dplyr)
# download US treasury 10Y constant-maturity
#   yield data and compute a total return series
t10_yield <- get_yields("DGS10")
t10_tr <- total_return(yields = t10_yield, maturity = 10)
head(t10_tr)

# step-by-step calculation
t10_yield <- get_yields("DGS10", format_out = "tibble")
t10_tr <- t10_yield %>%
  mutate(convexity = convexity(DGS10, 10),
         mod_duration = mod_duration(DGS10, 10),
         TR = total_return(DGS10, 10, mod_duration, convexity))
head(t10_tr)
```

`xts_to_tibble` *Convert an xts object to a tibble data frame*

Description

Convert an xts object to a tibble data frame

Usage

```
xts_to_tibble(x)
```

Arguments

`x`	an xts object

Value

A tibble data frame with first column 'date'

Note

If this function is used outside of the package's functions, i.e. using other data than FRED's, then make sure that the tibble does not contain non-numeric columns. An xts object is a matrix with an index attribute (date) and one cannot mix types in a matrix.

Index

convexity, 2

get_yields, 3

mod_duration, 4

tibble_to_xts, 5
total_return, 5

xts_to_tibble, 7