Package ‘smqf’

August 30, 2026

Title Statistical Methods for Quantitative Finance

Version 1.1-8

Description
Provides data and functions used in the book ``Statistical Methods for Quantitative Finance'' by David Ardia (2026).

License GPL-3

LazyData true

LazyDataCompression xz

Encoding UTF-8

Depends R (>= 4.1.0), xts

URL https://github.com/ArdiaD/smqf-package

BugReports https://github.com/ArdiaD/smqf-package/issues

Suggests zoo, testthat (>= 3.0.0), PerformanceAnalytics, glmnet, quadprog, copula, Matrix, readxl, MASS, mvtnorm

Config/testthat/edition 3

Imports graphics, stats, nloptr, pracma

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author David Ardia [aut, cre, cph, fnd] (ORCID: <https://orcid.org/0000-0003-2823-782X>),
Marius Hofert [ctb, cph] (Original author of qrmdata, from which 15 market datasets were ported under GPL-2 | GPL-3),
Kurt Hornik [ctb, cph] (Original author of qrmdata, from which 15 market datasets were ported under GPL-2 | GPL-3),
Alexander J. McNeil [ctb, cph] (Original author of qrmdata, from which 15 market datasets were ported under GPL-2 | GPL-3),
Dries Cornilly [ctb, cph] (Original author of mvskPortfolios, from which f_mvsk_portfolio() was ported under GPL-2 | GPL-3),
Kris Boudt [ctb, cph] (Original author of mvskPortfolios, from which f_mvsk_portfolio() was ported under GPL-2 | GPL-3)

Maintainer David Ardia <david.ardia.ch@gmail.com>

Repository CRAN

Date/Publication 2026-08-30 14:10:02 UTC

Contents

CAC *CAC 40 Index (Daily, xts)*

Description

Daily adjusted close prices of the Cotation Assistée en Continu (CAC 40) stock index (ticker symbol `^FCHI`), from its first date of availability on Yahoo Finance to 2015-12-31.

Usage

```
data("CAC")
```

Format

An `xts` object with 6549 daily observations and a single column `^FCHI` containing adjusted close prices in index points. The time index spans from 1990-03-01 to 2015-12-31.

Details

Originally distributed as `CAC` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("CAC")
class(CAC)        # "xts" "zoo"
dim(CAC)
head(CAC)
```

DAX *DAX Index (Daily, xts)*

Description

Daily adjusted close prices of the Deutscher Aktienindex (DAX) stock index (ticker symbol `^GDAXI`), from its first date of availability on Yahoo Finance to 2015-12-30.

Usage

```
data("DAX")
```

Format

An `xts` object with 6355 daily observations and a single column `^GDAXI` containing adjusted close prices in index points. The time index spans from 1990-11-26 to 2015-12-30.

Details

Originally distributed as `DAX` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL (≥ 2) license as **qrmdata**.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("DAX")
class(DAX)        # "xts" "zoo"
dim(DAX)
head(DAX)
```

DJ *Dow Jones Industrial Average Index (Daily, xts)*

Description

Daily adjusted close prices of the Dow Jones Industrial Average (ticker symbol `^DJI`), from its first date of availability on Yahoo Finance to 2015-12-31.

Usage

```
data("DJ")
```

Format

An `xts` object with 7797 daily observations and a single column `^DJI` containing adjusted close prices in index points. The time index spans from 1985-01-29 to 2015-12-31.

Details

Originally distributed as `DJ` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("DJ")
class(DJ)        # "xts" "zoo"
dim(DJ)
head(DJ)
```

DJ_const *Dow Jones Industrial Average Constituents (Daily, xts)*

Description

Daily adjusted close prices for the 30 constituents of the Dow Jones Industrial Average as of 2016-01-03.

Usage

```
data("DJ_const")
```

Format

An `xts` object with 13595 daily observations and 30 columns, one per constituent (e.g., `AAPL`, `IBM`, `JPM`, `XOM`). The time index spans from 1962-01-02 to 2015-12-31.

Details

Originally distributed as `DJ_const` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL (≥ 2) license as **qrmdata**.

Missing values

The vast majority of the 101,285 `NA`s are leading: a constituent is `NA` on every date before it first traded. A small number, however, occur *after* a series has started, so trimming the leading block does not guarantee a complete matrix. In this dataset there are 25 such interior `NA`s, confined to three dates (1985-09-27, when the NYSE was closed; 1981-08-10; and 1983-09-23) and spread over 24 of the 30 columns.

Handle both cases explicitly, for example with `stats::cor(x, use = "pairwise.complete.obs")`, or by restricting to rows with `rowSums(is.na(x)) == 0`. See `EURSTX_const`, where interior `NA`s are far more common.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("DJ_const")
class(DJ_const)        # "xts" "zoo"
dim(DJ_const)
head(colnames(DJ_const))
```

`EURSTOXX` *Euro Stoxx 50 Index (Daily, xts)*

Description

Daily adjusted close prices of the Euro Stoxx 50 stock index (ticker symbol `^STOXX50E`), from its first date of availability on Yahoo Finance to 2015-12-23.

Usage

```
data("EURSTOXX")
```

Format

An `xts` object with 7445 daily observations and a single column `^STOXX50E` containing adjusted close prices in index points. The time index spans from 1986-12-31 to 2015-12-23.

Details

Originally distributed as `EURSTOXX` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("EURSTOXX")
class(EURSTOXX)        # "xts" "zoo"
dim(EURSTOXX)
head(EURSTOXX)
```

EURSTX_const *Euro Stoxx 50 Constituents (Daily, xts)*

Description

Daily adjusted close prices for the 50 constituents of the Euro Stoxx 50 stock index as of 2016-01-03.

Usage

```
data("EURSTX_const")
```

Format

An `xts` object with 4174 daily observations and 50 columns, one per constituent (e.g., `SAP.DE`, `BNP.PA`, `SAN.MC`). The time index spans from 2000-01-03 to 2015-12-31.

Details

Originally distributed as `EURSTX_const` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**.

Missing values

Of the 8,270 `NA`s, 1,938 occur *after* a series has started, and every one of the 50 columns is affected. Trimming the leading block is therefore **not** enough to obtain a complete matrix.

The interior `NA`s are mostly local trading holidays: the constituents are listed on exchanges with different calendars (Paris, Frankfurt, Madrid, Amsterdam, Milan, Brussels), so a day that is a holiday in one country still appears in the index with `NA` for that country's shares. They span 1,015 distinct dates, the largest being Good Friday 2011-04-22 and Easter Monday 2011-04-25 (48 series each), Labour Day 2006-05-01 (46) and Christmas 2009-12-25 (43). A few series also have long individual gaps, notably `UL.PA` with 679 interior `NA`s.

Consequently `stats::cov(x)` returns `NA`s on this dataset. Use `use = "pairwise.complete.obs"` and project the result onto the positive semidefinite cone with `as.matrix(Matrix::nearPD(S)$mat)`, or restrict to rows with `rowSums(is.na(x)) == 0` for a single common estimation window.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("EURSTX_const")
class(EURSTX_const)        # "xts" "zoo"
dim(EURSTX_const)
head(colnames(EURSTX_const))
```

---

extdata — *Raw Data Files Shipped with the Book's Import Examples*

Description

Chapter 1 of the book opens by reading a plain file from disk, so that readers practise importing data before they meet packaged datasets. Those files are installed with **smqf** under `extdata` and can be located with `system.file`.

Details

Copy them into a `data/` folder of your own project to follow the chapter exactly:

```
dir.create("data", showWarnings = FALSE)
file.copy(dir(system.file("extdata", package = "smqf"), full.names = TRUE),
          "data", overwrite = FALSE)
```

Files

`stocks.txt` Whitespace-delimited, header row, 574 weekly observations from 2008-01-01 to 2018-12-25. Columns `Date`, `AAPL`, `AXP`, `BA`, `CAT`, `CSCO`: split- and dividend-adjusted close prices in USD for five constituents of the Dow Jones Industrial Average. Read with `read.table(f, header = TRUE)`.

`stocks.csv` The same 574 x 6 table, comma-separated. Read with `read.csv()` or `readr::read_csv()`.

`stocks.xlsx` The same table as an Excel workbook, for the `readxl::read_excel()` example.

`prices.txt` Whitespace-delimited, 6,109 daily observations from 1990-02-01 to 2013-07-02, no missing values. Columns `Date`, `S&P 500`, `FTSE 100`, `CAC 40`, `DAX 30`: index levels in local currency. Not used in Chapter 1; supplied for open-ended exercises. Note the non-syntactic column names — read with `check.names = FALSE` to preserve them.

Weekly timestamp convention

The dates in `stocks.txt`, `stocks.csv` and `stocks.xlsx` are all Tuesdays, spaced exactly seven days apart, because the series was downloaded at weekly periodicity starting on 2008-01-01 (a Tuesday). The timestamp labels the week, **not** the day the price was observed: the price recorded against a Tuesday is the closing price at the end of that calendar week.

This matters whenever the series is combined with weekly data that is stamped at the end of the week, as the Fama–French factors in `FamaFrench` are (Fridays). The return computed for Tuesday t spans the week ending on the **Friday of that same calendar week**, so the two series pair one-to-one in order:

```
returns   2017-01-03  2017-01-10  2017-01-17   (Tuesday stamps)
factors   2017-01-06  2017-01-13  2017-01-20   (Friday stamps)
          |___________|___________|__________|  same weeks, paired in order
```

Do **not** align them by carrying observations forward across a merged union index (for instance with two `zoo::na.locf()` passes): because the two stamps interleave, every Tuesday would inherit the *preceding* Friday's factors and the whole factor series would lag the returns by one week. Match by position, or restamp one series onto the other's convention, and assert that the two lengths agree before regressing.

See Also

`FamaFrench` for the weekly factors these prices are regressed on in Chapter 1.

Examples

```
f <- system.file("extdata", "stocks.txt", package = "smqf")
stocks <- read.table(f, header = TRUE)
dim(stocks)
head(stocks, 3)
unique(weekdays(as.Date(stocks$Date)))   # all Tuesdays
```

`FamaFrench` *Fama–French Factors (Weekly, xts)*

Description

Weekly time series of the three Fama–French equity risk factors and the risk-free rate, as provided by Kenneth French's data library. Values are expressed in **percentage points** (e.g., `1.60` means a return of 1.60%).

Usage

```
data("FamaFrench")
```

Format

An `xts` object with 4\,834 weekly observations (from 1926-07-02 to 2019-02-22) and 4 columns:

mkt_rf Excess return on the market (market return minus risk-free rate), in %.

smb Small-Minus-Big size factor return, in %.

hml High-Minus-Low value factor return, in %.

rf Risk-free rate (weekly), in %.

Details

The time index is a weekly `Date`, stamped at the **end** of each week. In the modern sample that is a Friday, shifted back to the Thursday when the Friday is a holiday (3,535 Fridays and 137 Thursdays in total). In the early history, when Saturday trading sessions were still held, the stamp is a Saturday: 1,158 observations, running to the late 1950s. Do not assume a Friday stamp throughout. Divide by 100 to convert to decimal returns.

Note also that `rf` is slightly negative in 60 Depression-era weeks (1933 and 1938), with a minimum of -0.016%. That is a property of the source data, not an error.

Aligning with other weekly series

Weekly price series downloaded from vendors are often stamped at the *start* of the week instead, so the two indices interleave rather than coincide — see `extdata` for the prices used in Chapter 1, which carry Tuesday stamps. Merging such a series with these factors and then filling the gaps, for instance with two `zoo::na.locf()` passes, silently lags the factors by one week and can flip the sign of an estimated market beta. Pair the two series by position over a common span, or restamp one onto the other's convention, and assert that the lengths match before regressing.

Downsampling with `FamaFrench[xts::endpoints(FamaFrench, "months"), ]` selects the *last weekly observation* of each month; it does not compound the weeks into a monthly return. To obtain monthly factor returns, aggregate the weekly ones explicitly.

Source

Kenneth R. French Data Library, https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

References

Fama, E. F., & French, K. R. (1993). Common Risk Factors in the Returns on Stocks and Bonds. *Journal of Financial Economics*, 33(1), 3–56.

Examples

```
data("FamaFrench")
class(FamaFrench)        # "xts" "zoo"
dim(FamaFrench)          # 4834 x 4
head(FamaFrench)

# Convert to decimal and extract the three equity factors
ff <- FamaFrench[, c("mkt_rf", "smb", "hml")] / 100
rf <- FamaFrench[, "rf"] / 100
```

Fred *FRED-MD Macro Factors and Dow Jones Index Changes (Monthly, 2015–2019)*

Description

An `xts` object holding 128 FRED-MD macroeconomic predictors and the one-month-ahead change in the Dow Jones Industrial Average, aligned over the period 2015-01 to 2019-12. Used to illustrate high-dimensional regularised regression (Lasso, Ridge) in a prediction context with more predictors than observations.

Usage

```
data("Fred")
```

Format

An `xts` object with 60 monthly observations (index Jan 2015 to Dec 2019, class `zoo::yearmon`) and 129 columns:

columns 1 to 128 FRED-MD macro variables, transformed (differenced or log-differenced) to achieve stationarity following the FRED-MD transformation codes. Column names are FRED-MD series codes (e.g., `"INDPRO"`, `"CPIAUCSL"`, `"GS10"`). The series are **not** rescaled: each keeps the units of its own transform, and the column standard deviations span roughly eight orders of magnitude, from `CONSPI` (about 0.0009) to `BOGMBASE` (about 73,000). Any penalized fit must therefore standardize — which `glmnet::glmnet()` does by default, but only while `standardize = TRUE`.

`DJI.Adjusted` **(final column)** One-month-ahead change in the Dow Jones Industrial Average, aligned so that row t of the predictors corresponds to row t of the response. **This is a difference in index points, not a return**: it ranges from about -2211 to $+1785$ with a standard deviation near 780. Because the index level roughly doubles over the sample, the series is heteroskedastic by construction. Divide by the lagged index level before interpreting any result as a return.

Details

The response is a difference in *index points*, not a return. See the `DJI.Adjusted` entry under Format before interpreting any coefficient.

The 128 macro predictors (columns 1 to 128) were downloaded from the McCracken–Ng FRED-MD database and transformed according to the recommended stationarity codes. The target column `DJI.Adjusted` was obtained from Yahoo Finance via `quantmod::getSymbols("^DJI")` and differenced one month ahead, giving a change in index points rather than a return: its standard deviation is about 780 and it ranges from roughly -2211 to +1785. All series are restricted to the 60-month window 2015-01 to 2019-12 and temporally aligned so that the predictor columns in row t can be used to predict `DJI.Adjusted` in row t.

Source

- McCracken, M. W., & Ng, S. FRED-MD database, https://www.stlouisfed.org/research/economists/mccracken/fred-databases.
- Dow Jones daily prices via `quantmod::getSymbols("^DJI")`, Yahoo Finance.

References

McCracken, M. W., & Ng, S. (2016). FRED-MD: A Monthly Database for Macroeconomic Research. *Journal of Business & Economic Statistics*, 34(4), 574–589.

See Also

The book chapter on *Dimension Reduction* (Chapter 4 of *Statistical Methods for Quantitative Finance*) introduces Lasso and Ridge regression on the lower-dimensional `GoyalWelch` dataset; `Fred` provides a complementary high-dimensional ($p \gg n$) test bed for the same regularised-regression workflow.

Examples

```
data("Fred")
class(Fred)                                  # "xts" "zoo"
y <- Fred[, "DJI.Adjusted"]                  # response (60 x 1)
X <- Fred[, colnames(Fred) != "DJI.Adjusted"]  # 128 macro predictors
dim(X)

# Lasso with cross-validation (requires glmnet, which expects matrices)
if (requireNamespace("glmnet", quietly = TRUE)) {
  set.seed(1234)
  fit <- glmnet::cv.glmnet(as.matrix(X), as.numeric(y), alpha = 1)
  coef(fit, s = "lambda.min")[coef(fit, s = "lambda.min")[,1] != 0, , drop = FALSE]
}
```

`FTSE` *FTSE 100 Index (Daily, xts)*

Description

Daily adjusted close prices of the FTSE 100 stock index (ticker symbol `^FTSE`), from its first date of availability on Yahoo Finance to 2015-12-31.

Usage

```
data("FTSE")
```

Format

An `xts` object with 8333 daily observations and a single column `^FTSE` containing adjusted close prices in index points. The time index spans from 1984-01-03 to 2015-12-31.

Details

Originally distributed as FTSE in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("FTSE")
class(FTSE)        # "xts" "zoo"
dim(FTSE)
head(FTSE)
```

`FTSE_const` *FTSE 100 Constituents (Daily, xts)*

Description

Daily adjusted close prices for 98 constituents of the FTSE 100 stock index as of 2016-01-03.

Usage

```
data("FTSE_const")
```

Format

An `xts` object with 7198 daily observations and 98 columns, one per constituent (e.g., `HSBA.L`, `BP.L`, `VOD.L`). The time index spans from 1988-05-03 to 2015-12-31.

Details

Originally distributed as `FTSE_const` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**. Only 98 of the 100 constituents were available at the time of the download.

Missing values

Most of the 163,821 NAs are leading: a constituent is NA on every date before it first traded. But 4,634 of them occur *after* a series has started, spread over 96 of the 98 columns, so trimming the leading block does not produce a complete matrix.

Handle both cases explicitly. Two idioms cover most needs: restrict to the columns that are complete over your window, as the book does with `x <- x[, colSums(is.na(x)) == 0]` — over 2013–2015 this keeps 95 of the 98 names — or estimate pairwise with `stats::cov(x, use = "pairwise.complete.obs")` and project the result onto the positive semidefinite cone. See `EURSTX_const`, where interior NAs are far more common, and `DJ_const`, where they are rare.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("FTSE_const")
class(FTSE_const)        # "xts" "zoo"
dim(FTSE_const)
head(colnames(FTSE_const))
```

FungHsieh — *Fung–Hsieh Factors (Monthly, xts)*

Description

Monthly time series of commonly used Fung–Hsieh style and macro factors. The object is an `xts` matrix indexed by month (class `zoo::yearmon`) with the columns listed below. Typical use cases include hedge-fund replication and factor attribution.

Usage

```
data("FungHsieh")
```

Format

An `xts` object with 276 monthly observations (Jan 1994 to Dec 2016), indexed by `zoo::yearmon`, and 8 columns.

All columns are expressed in percentage points, never in basis points or decimals. Divide by 100 before combining them with decimal return series such as `PerformanceAnalytics::edhec`.

EMKT Equity market factor, % per month. Mean 0.62, sd 4.24 (14.7% annualized).

RF Risk-free rate, % per month. Always non-negative; maximum 0.56, i.e. 6.7% annualized.

SS Size spread (small minus large), % per month. Mean 0.05, sd 3.26, correlation with EMKT of 0.08 — the near-zero mean and low market correlation confirm a long–short spread rather than a directional return.

CST10Y *Change* in the 10-year U.S. Treasury constant-maturity yield, in percentage points per month (range -1.11 to 0.65). Not a return.

BAA *Change* in the Moody's Baa credit spread, in percentage points per month (range -0.99 to 1.45). Not a return.

PTFSBD Fung–Hsieh primitive trend-following straddle: bond, % per month. Mean -1.67, sd 15.23.

PTFSCOM Fung–Hsieh primitive trend-following straddle: commodity, % per month. Mean -0.55, sd 14.28.

PTFSFX Fung–Hsieh primitive trend-following straddle: currency, % per month. Mean -0.85, sd 19.48.

Details

The three PTFS* series are lookback-straddle returns. They are strongly right-skewed with large positive tails (up to +89.8% in a single month) and negative means, which is the expected signature of a long-option payoff, not a data error.

Two of the eight columns — CST10Y and BAA — are changes in yields rather than returns. Do not rescale them as if they were returns, and interpret their loadings as sensitivities per percentage point of yield change.

The time index is of class zoo::yearmon (a calendar month with no day-of-month), so the series aligns with other monthly data by month, irrespective of any day-of-month convention. To merge with a Date-indexed series, coerce its index with as.yearmon().

Provenance

This object is a static snapshot compiled for the book's examples, not a maintained feed, and the limits of what is documented about it should be stated plainly. The original download URLs and the vintage of each input series were not recorded when the snapshot was taken, so data-raw/FungHsieh.R is a *contract harness* rather than a reconstruction script: it pins every property the book relies on and will fail loudly if a future rebuild diverges, but it cannot regenerate the object from source.

The consequence is that the column semantics described above are *inferred* from the behaviour of the shipped series and from the standard Fung–Hsieh construction, not verified against an archived input file. Some of them are corroborated by the data: BAA correlates -0.51 with CST10Y and -0.27 with EMKT, which is the signature of a credit *spread* change (spreads widen as equities fall and as safe yields drop) and rules out a raw Baa yield change, which would track the ten-year yield positively. Others are not: the exact index pair underlying SS, the day-of-month sampling convention behind CST10Y, and the precise straddle construction behind the PTFS* series cannot be settled from the object alone.

Treat the factor identifications as adequate for the illustrative regressions in the book, and re-derive the factors from https://people.duke.edu/~dah7/HFRFData.htm before publishing any attribution result that turns on the definition of a particular factor.

Source

Compiled from public factor sources commonly used in the Fung–Hsieh literature: David A. Hsieh's hedge-fund data library (https://people.duke.edu/~dah7/HFRFData.htm) for the PTFS* straddle factors, and FRED/H.15 for the yield and credit series. See data-raw/FungHsieh.R for the assembly steps.

References

Fung, W., & Hsieh, D. A. (2004). Hedge Fund Benchmarks: A Risk-Based Approach. *Financial Analysts Journal*, 60(5), 65–80. Fung, W., & Hsieh, D. A. (2001). The Risk in Hedge Fund Strategies: Theory and Evidence from Trend Followers. *Review of Financial Studies*, 14(2), 313–341.

Examples

```
data("FungHsieh")
class(FungHsieh)              # "xts" "zoo"
head(FungHsieh)
colnames(FungHsieh)
# Quick plot of the three PTFS factors
if (requireNamespace("zoo", quietly = TRUE)) {
  zoo::plot.zoo(FungHsieh[, c("PTFSBD", "PTFSCOM", "PTFSFX")], screens = 1, col = 1:3)
}
```

---

f_clayton_copula_2d_pdf

Clayton Copula PDF (2-Dimensional)

Description

Computes the value of the bivariate Clayton copula probability density function (PDF) at a specified point (u_1, u_2) for a given dependence parameter θ.

Usage

```
f_clayton_copula_2d_pdf(u, theta, log = FALSE)
```

Arguments

u	Numeric vector of length 2, containing values in the interval $(0, 1]$ representing the evaluation point (u_1, u_2).
theta	Numeric scalar giving the dependence parameter
log	Logical: return the log-density instead of the density. The Clayton density diverges towards the lower corner, where only the log scale is representable. Default FALSE. ($\theta \geq 0$). The value $\theta = 0$ is the independence limit.

Details

The Clayton copula is an Archimedean copula with lower-tail dependence $\lambda_L = 2^{-1/\theta}$ and no upper-tail dependence ($\lambda_U = 0$). When $\theta = 0$, the copula reduces to the independence copula, whose PDF equals 1 for all $(u_1, u_2) \in [0, 1]^2$.

The PDF is given by:

$$c(u_1, u_2; \theta) = (1 + \theta)\,(u_1 u_2)^{-(1+\theta)}\,\left(u_1^{-\theta} + u_2^{-\theta} - 1\right)^{-2-1/\theta}.$$

Numerical computation may lose precision for large θ (typically above about 34).

Value

A numeric value corresponding to the Clayton copula PDF evaluated at the specified point.

References

- Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall. - Patton, A. (2006). Modelling Asymmetric Exchange Rate Dependence. *International Economic Review*, 47(2), 527–556.

See Also

f_gumbel_copula_2d_pdf, f_normal_copula_pdf, f_student_copula_pdf

Examples

```
# Example: Evaluate Clayton copula PDF at (u1, u2) = (0.5, 0.5)
f_clayton_copula_2d_pdf(c(0.5, 0.5), theta = 2)
```

---

f_display_copula *Display a Bivariate Copula Surface*

Description

Evaluates and displays a bivariate copula function (typically a PDF or CDF) on a 2D grid, producing a static 3D surface via base R's `persp()`.

Usage

```
f_display_copula(my_copula, grid_1, grid_2, plot = TRUE, zlab = "Pdf", ...)
```

Arguments

`my_copula`	A function that takes a numeric vector `c(u1, u2)` as input and returns a scalar value (e.g., a copula PDF or CDF).
`grid_1, grid_2`	Numeric vectors defining the evaluation grid for u_1 and u_2 in the interval $[0, 1]$.
`plot`	Logical; if `TRUE` (the default), a 3D surface is drawn via `graphics::persp()`. If `FALSE`, no graphics device is opened and only the evaluated matrix is returned (useful for testing or non-interactive use).
`zlab`	Character string used as the vertical-axis label of the `persp()` plot. Defaults to `"Pdf"`; set e.g. `zlab = "Cdf"` when displaying a copula CDF.
`...`	Additional arguments passed to `graphics::persp()` (e.g., `theta`, `phi`, `col`).

Details

This function provides a simple way to visualize the surface of a bivariate copula (e.g., Clayton, Gumbel, Gaussian). The copula function should accept a vector of two uniform values (u_1, u_2) and return its density or CDF value.

Value

Invisibly returns the matrix of evaluated copula values `f_U`, with rows corresponding to `grid_1` and columns to `grid_2`.

Examples

```
# Example: Display the Clayton copula PDF surface
grid <- seq(0.05, 0.95, length.out = 25)
f_display_copula(
  my_copula = function(u) f_clayton_copula_2d_pdf(u, theta = 2),
  grid_1 = grid,
  grid_2 = grid
)
```

`f_efficient_frontier` *Compute a Long-Only Mean–Variance Efficient Frontier*

Description

Solves a sequence of quadratic programs to trace the long-only, fully-invested Markowitz efficient frontier between the minimum-variance portfolio and the maximum-return portfolio.

Usage

```
f_efficient_frontier(mu, Sigma, n_ptf, targets = NULL)
```

Arguments

mu
: Numeric vector of length N: expected asset returns.

Sigma
: Numeric $N \times N$ covariance matrix. Objects that are not base matrices (for instance the `dpoMatrix` returned by `Matrix::nearPD()$mat`) are coerced with `as.matrix()`. Only the symmetric part is used internally.

n_ptf
: Integer ≥ 2: number of portfolios along the frontier (including the minimum-variance and maximum-return portfolios).

targets
: Optional numeric vector of target returns, sorted increasingly. When supplied, the frontier is solved at exactly these targets instead of the default grid, and `n_ptf` must be omitted or equal to `length(targets)`. This is what makes several frontiers comparable: frontiers computed on different inputs otherwise use different grids, and averaging them point by point mixes portfolios that target different returns. Targets above the maximum attainable return are reported with status `"failed"` rather than silently clipped.

Details

Portfolios are obtained by minimizing variance for a grid of target returns under the constraints $\sum_i w_i = 1$ and $w_i \geq 0$.

Each QP solves

$$\min_w \; w^\top \Sigma \, w \quad \text{s.t. } \mathbf{1}^\top w = 1, \; w \geq 0$$

and, for every point beyond the minimum-variance portfolio, additionally

$$\mu^\top w = \mu^\star,$$

where $\mu^\star$ spans a linear grid between the min-variance portfolio return and $\max(\mu)$.

The maximum-return end point is obtained from the same constrained QP rather than by assigning weights by hand. This matters when several assets tie for `max(mu)`: spreading weight equally across them is feasible but generally not variance-minimal, so the resulting portfolio can be strictly dominated and would not lie on the frontier.

`Sigma` is checked for symmetry and positive semidefiniteness. A matrix whose smallest eigenvalue falls clearly below zero is rejected rather than silently repaired; project it first, for example with `as.matrix(Matrix::nearPD(Sigma)$mat)`.

If a QP fails to converge, the corresponding point is returned as `NA` with `status == "failed"` and a warning, instead of being filled with a neighbouring solution that does not meet its target return.

Requires **pracma** for `quadprog`.

Value

A list with components:

weights Numeric matrix $N \times n_ptf$: portfolio weights. Rows are named after `mu` when it carries names; columns are labelled `ptf1`, `ptf2`, ... Columns of failed solves are `NA`.

volatility Named numeric of length n_ptf: portfolio standard deviations, `NA` where the solve failed.

expected_returns Named numeric of length n_ptf: portfolio expected returns, `NA` where the solve failed.

frontier Data frame with columns `point` (integer index), `expected_return`, `volatility`, `target` and `status` for each frontier portfolio.

targets Numeric vector of length n_ptf: the target-return grid used to trace the frontier.

status Character vector of length n_ptf: `"ok"` for a solved point, `"failed"` otherwise.

References

Markowitz, H. (1952). Portfolio Selection. *Journal of Finance*, 7(1), 77–91.

See Also

`f_ptf_max_U`, `f_portfolio_moments`, `quadprog`

Examples

```
set.seed(1)
N <- 4
mu <- c(0.08, 0.10, 0.12, 0.09)
M  <- matrix(rnorm(N*N), N); Sigma <- crossprod(M) / N  # PSD covariance
ef <- f_efficient_frontier(mu, Sigma, n_ptf = 20)
# Inspect end points
ef$expected_returns[c(1, 20)]
ef$volatility[c(1, 20)]
all(ef$status == "ok")

# Ties for the maximum expected return are resolved by minimizing variance,
# not by equal weighting: assets 2 and 3 both reach 0.2, and the variance-
# minimal mix is c(0, 1/101, 100/101) with variance 100/101, well below the
# 25.25 of an equal split and below the 1 of asset 3 on its own.
ef_tie <- f_efficient_frontier(mu = c(0.1, 0.2, 0.2),
                               Sigma = diag(c(1, 100, 1)), n_ptf = 5)
ef_tie$weights[, 5]

# A common target grid makes two frontiers comparable point by point
mu2 <- mu + c(0.01, -0.01, 0.00, 0.02)
grid <- seq(0.095, 0.105, length.out = 5)
e1 <- f_efficient_frontier(mu,  Sigma, targets = grid)
e2 <- f_efficient_frontier(mu2, Sigma, targets = grid)
# Same targets, so the two weight vectors at point 3 are directly comparable
cbind(e1$weights[, 3], e2$weights[, 3])
```

`f_gumbel_copula_2d_cdf` *Gumbel Copula CDF (2-Dimensional)*

Description

Computes the bivariate Gumbel copula cumulative distribution function (CDF) at (u_1, u_2) for dependence parameter $\theta \geq 1$.

Usage

```
f_gumbel_copula_2d_cdf(u, theta)
```

Arguments

u	Numeric vector of length 2 with entries in $(0, 1]$: the evaluation point (u_1, u_2).
theta	Numeric scalar, the Gumbel dependence parameter $\theta \geq 1$.

Details

The Gumbel copula exhibits upper-tail dependence $\lambda_U = 2 - 2^{1/\theta}$ and no lower-tail dependence ($\lambda_L = 0$). When $\theta = 1$, it reduces to the independence copula with $C(u_1, u_2) = u_1 u_2$.

The CDF is

$$C(u_1, u_2; \theta) = \exp\Big(- \big[(-\log u_1)^\theta + (-\log u_2)^\theta\big]^{1/\theta}\Big), \qquad \theta \geq 1.$$

Value

A numeric scalar: $C(u_1, u_2; \theta)$.

References

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall. Nelsen, R. B. (2006). *An Introduction to Copulas* (2nd ed.). Springer.

See Also

f_gumbel_copula_2d_pdf, f_clayton_copula_2d_pdf, f_normal_copula_pdf, f_student_copula_pdf

Examples

```
f_gumbel_copula_2d_cdf(c(0.5, 0.8), theta = 2)
f_gumbel_copula_2d_cdf(c(0.7, 0.7), theta = 1)  # independence: ~0.49
```

---

f_gumbel_copula_2d_pdf

Gumbel Copula PDF (2-Dimensional)

Description

Computes the bivariate Gumbel copula probability density function (PDF) at (u_1, u_2) for dependence parameter $\theta \geq 1$.

Usage

```
f_gumbel_copula_2d_pdf(u, theta)
```

Arguments

`u`	Numeric vector of length 2 with entries in the open interval $(0, 1)$: the evaluation point (u_1, u_2). The boundary value 1 is excluded because the density is not finite at the corner $(1, 1)$ for $\theta > 1$ (upper-tail dependence).
`theta`	Numeric scalar, the Gumbel dependence parameter $\theta \geq 1$.

Details

The PDF can be written as

$$c(u_1, u_2; \theta) = C(u_1, u_2; \theta) \frac{\{[-\log u_1]\ [-\log u_2]\}^{\theta-1}}{u_1 u_2} \left((-\log u_1)^\theta + (-\log u_2)^\theta\right)^{1/\theta-2} \left(\{(-\log u_1)^\theta + (-\log u_2)^\theta\}^{1}\right.$$

When $\theta = 1$, $c(u_1, u_2; 1) \equiv 1$ (independence).

Value

A numeric scalar: $c(u_1, u_2; \theta)$.

Note

Numerical stability may degrade as $u_i \to 0^+$ (large $-\log u_i$).

References

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall. Nelsen, R. B. (2006). *An Introduction to Copulas* (2nd ed.). Springer.

See Also

`f_gumbel_copula_2d_cdf`, `f_clayton_copula_2d_pdf`, `f_normal_copula_pdf`, `f_student_copula_pdf`

Examples

```
f_gumbel_copula_2d_pdf(c(0.5, 0.8), theta = 2)
f_gumbel_copula_2d_pdf(c(0.7, 0.7), theta = 1)  # independence: 1
```

`f_mdp`	*Most Diversified Portfolio*

Description

Compute the weights of the most diversified portfolio (MDP) of Choueifaty and Coignard (2008), which maximizes the diversification ratio

$$DR(w) = \frac{w'\sigma}{\sqrt{w'\Sigma w}}\,,$$

where $\sigma = (\sqrt{\Sigma_{11}}, \ldots, \sqrt{\Sigma_{dd}})'$ is the vector of asset volatilities. Without further constraints the solution is available in closed form, $w \propto \Sigma^{-1}\sigma$, normalized to sum to one.

Usage

```
f_mdp(Sigma, tol = 1e-08)
```

Arguments

`Sigma`	A $d \times d$ covariance matrix. Must be symmetric and positive definite: the closed form requires an inverse, so a singular `Sigma` (for instance a sample covariance estimated from fewer observations than assets) is rejected rather than silently solved.
`tol`	Numeric tolerance used to check symmetry and positive definiteness. Default `1e-8`.

Details

The solution is **unconstrained beyond the budget constraint**: weights may be negative, and in large universes typically are, often substantially. A long-only most diversified portfolio is a different object and requires a constrained quadratic program; it is not what this function returns. Inspect `sum(abs(w))` to see the gross exposure implied by the solution.

The diversification ratio is scale invariant in w, so the normalization $\sum_i w_i = 1$ is a choice of representative and does not change the portfolio.

Value

A named numeric vector of d portfolio weights summing to one. Names are taken from `dimnames(Sigma)` when available.

References

Choueifaty, Y., & Coignard, Y. (2008). Toward Maximum Diversification. *Journal of Portfolio Management*, 35(1), 40–51.

See Also

`f_efficient_frontier`, `f_ptf_max_U`

Examples

```
Sigma <- matrix(c(0.04, 0.01, 0.00,
                  0.01, 0.09, 0.02,
                  0.00, 0.02, 0.16), 3, 3,
                dimnames = list(c("A", "B", "C"), c("A", "B", "C")))
w <- f_mdp(Sigma)
w
sum(w)

# Diversification ratio of the MDP against the equally weighted portfolio
f_dr <- function(w, Sigma) {
  sum(w * sqrt(diag(Sigma))) / sqrt(drop(crossprod(w, Sigma %*% w)))
}
c(mdp = f_dr(w, Sigma), equal = f_dr(rep(1 / 3, 3), Sigma))
```

`f_mvsk_portfolio` *Higher-Moment Efficient Portfolios*

Description

Tilts a reference portfolio towards more preferable moments — lower variance, higher skewness, lower kurtosis and optionally higher mean — while allowing the reference criterion to deteriorate by at most a margin `kappa`. This implements the algorithmic tilting of Boudt, Cornilly, Van Holle & Willems (2020).

Usage

```
f_mvsk_portfolio(
  m1 = NULL,
  M2 = NULL,
  M3 = NULL,
  M4 = NULL,
  w0 = NULL,
  g = NULL,
  lb = NULL,
  ub = NULL,
  lin_eq = NULL,
  lin_eqC = NULL,
  nlin_eq = NULL,
  lin_ieq = NULL,
  lin_ieqC = NULL,
  nlin_ieq = NULL,
  href = NULL,
  kappa = NULL,
  relative = FALSE,
  param = NULL,
  options = list(),
  mompref = NULL
)
```

Arguments

`m1`	Numeric vector of expected returns, or `NULL` to leave the mean out of the tilting.
`M2`	Numeric $d \times d$ covariance matrix.
`M3`	Numeric $d \times d^2$ coskewness matrix, as returned by `PerformanceAnalytics::M3.MM()`, or `NULL`.
`M4`	Numeric $d \times d^3$ cokurtosis matrix, as returned by `PerformanceAnalytics::M4.MM()`, or `NULL`.

`w0`	Reference portfolio: either a numeric weight vector, or the name of a criterion to solve for it — `"DR"` (most diversified), `"ERC"` (equal risk contribution), `"HI"` / `"EW"` (equally weighted), `"EU"` (expected utility, needs `param$gamma`) or `"TEvol"`. Defaults to equal weights.
`g`	Direction of moment improvement: a numeric vector, a function of `delta`, or one of `"mvsk"` / `"vsk"` to derive it from the reference portfolio's own moments.
`lb, ub`	Lower and upper bounds for the weights. Default `0` and `1`.
`lin_eq, lin_eqC`	Linear equality constraints $Aw = b$. Default is the full-investment constraint.
`nlin_eq`	Function returning non-linear equality constraints and their Jacobian.
`lin_ieq, lin_ieqC`	Linear inequality constraints $Aw \leq b$.
`nlin_ieq`	Function returning non-linear inequality constraints and their Jacobian.
`href`	Reference criterion held in check while tilting: a name (as for `w0`) or a function.
`kappa`	Numeric vector of margins by which `href` may deteriorate.
`relative`	Logical: is `kappa` relative rather than absolute?
`param`	List of extra arguments for the `href` function, such as `gamma` for `"EU"` or `wref` for `"TEvol"`.
`options`	List of **nloptr** options.
`mompref`	Direction of preference per moment (`+1` higher, `-1` lower). Defaults to higher mean and skewness, lower variance and kurtosis.

Details

The co-moment matrices must be in the **matrix** form returned by `PerformanceAnalytics::M3.MM(x)` and `M4.MM(x)` with the default `as.mat = TRUE`. If you hold the compact vector form, convert it first with `PerformanceAnalytics::M3.vec2mat()` and `M4.vec2mat()`.

`delta` is the largest improvement attainable within the margin. It is non-decreasing in `kappa` — a wider margin never shrinks the feasible set — but it need not increase strictly, and at `kappa = 0` it is zero only up to solver tolerance.

Value

A list with the tilted `weights` (one row per `kappa`), the achieved improvement `delta`, the resulting portfolio `moms`, the constraint values `constr`, and the solver `status` for each margin.

Provenance

This function is a port of `mvskPortfolio()` from the **mvskPortfolios** package by Dries Cornilly and Kris Boudt (https://github.com/c3jg/mvskPortfolios), redistributed here under the same GPL ($\geq$ 2) licence so that the book's examples install from CRAN alone. The algorithm is unchanged; the internal calls to non-exported **PerformanceAnalytics** symbols have been replaced by the equivalent helpers of this package, and the solver status is now reported.

Author(s)

Dries Cornilly and Kris Boudt, ported by David Ardia.

References

Boudt, K., Cornilly, D., Van Holle, F., & Willems, J. (2020). Algorithmic portfolio tilting to harvest higher moment gains. *Heliyon*, 6(3).

Briec, W., Kerstens, K., & Jokung, O. (2007). Mean-variance-skewness portfolio performance gauging: a general shortage function and dual approach. *Management Science*, 53(1), 135–149.

Choueifaty, Y., & Coignard, Y. (2008). Toward maximum diversification. *Journal of Portfolio Management*, 35(1), 40–51.

See Also

f_ptf_max_U, f_portfolio_moments

Examples

```
if (requireNamespace("PerformanceAnalytics", quietly = TRUE)) {
  data(edhec, package = "PerformanceAnalytics")
  x  <- edhec[, 1:5]
  M2 <- stats::cov(x)
  M3 <- PerformanceAnalytics::M3.MM(x)
  M4 <- PerformanceAnalytics::M4.MM(x)

  res <- f_mvsk_portfolio(M2 = M2, M3 = M3, M4 = M4, w0 = "DR",
                          g = "vsk", href = "DR",
                          kappa = c(0, 0.005, 0.01))
  round(100 * res$weights, 2)
  res$delta
}
```

---

f_normal_copula_pdf *Multivariate Normal Copula PDF*

Description

Computes the probability density function (PDF) of the Gaussian (normal) copula at a specified point $u \in [0,1]^N$, given mean vector μ and covariance matrix Σ of the underlying multivariate normal distribution.

Usage

```
f_normal_copula_pdf(u, mu, Sigma, log = FALSE)
```

Arguments

u	Numeric vector of length N with entries in $(0,1)$: the copula evaluation point.
mu	Numeric vector of length N, the mean of the corresponding multivariate normal distribution (usually zeros for a copula).

`Sigma`	Numeric positive-definite $N \times N$ covariance matrix.
`log`	Logical; if `TRUE` the log-density is returned. Defaults to `FALSE`. In high dimension the density itself may overflow or underflow the double-precision range even though its logarithm is perfectly well behaved, so `log = TRUE` is the safe choice for likelihoods.

Details

The Gaussian copula density is

$$c(u; \mu, \Sigma) = \frac{\phi_N\left(\Phi^{-1}(u); \mu, \Sigma\right)}{\prod_{i=1}^{N} \phi_1(\Phi^{-1}(u_i); \mu_i, \sigma_i^2)},$$

where ϕ_N and ϕ_1 are multivariate and univariate normal densities respectively, and Φ^{-1} denotes the inverse normal CDF applied componentwise. The resulting function is a valid copula density on the unit hypercube $[0, 1]^N$.

Typically, the copula is defined for $\mu = 0$ and correlation matrix R, but the implementation here generalizes to arbitrary mean and covariance.

Value

A numeric scalar: the value of the Gaussian copula density $c(u; \mu, \Sigma)$ at the point `u`, or its logarithm when `log = TRUE`. The value is returned as a plain numeric scalar (not a 1×1 matrix).

References

Joe, H. (1997). *Multivariate Models and Dependence Concepts.* Chapman & Hall. Nelsen, R. B. (2006). *An Introduction to Copulas* (2nd ed.). Springer. McNeil, A. J., Frey, R., & Embrechts, P. (2015). *Quantitative Risk Management.* Princeton University Press.

See Also

`f_student_copula_pdf`, `f_clayton_copula_2d_pdf`, `f_gumbel_copula_2d_pdf`

Examples

```
# Example: 2D Gaussian copula
Sigma <- matrix(c(1, 0.7, 0.7, 1), 2, 2)
mu <- c(0, 0)
f_normal_copula_pdf(c(0.5, 0.8), mu, Sigma)

# Compare with independence (Sigma = I)
f_normal_copula_pdf(c(0.5, 0.8), mu, diag(2))

# The independence copula has density 1 everywhere, in any dimension
d <- 300
f_normal_copula_pdf(rep(0.99, d), rep(0, d), diag(d))
```

f_portfolio_moments *Portfolio Co-Moments (Variance, Co-Skewness, Co-Kurtosis)*

Description

Compute the portfolio's centred 2nd, 3rd and 4th moments (variance, co-skewness, co-kurtosis) from the asset-level co-moment matrices and a weight vector.

Usage

```
f_portfolio_moments(w, M2 = NULL, M3 = NULL, M4 = NULL)
```

Arguments

`w`	Numeric vector of length d of portfolio weights.
`M2`	Optional $d \times d$ covariance matrix of the assets (as returned by `PerformanceAnalytics::M2.MM()`). If `NULL`, the second moment is not computed.
`M3`	Optional $d \times d^2$ co-skewness matrix (`PerformanceAnalytics::M3.MM()`). If `NULL`, the third moment is not computed.
`M4`	Optional $d \times d^3$ co-kurtosis matrix (`PerformanceAnalytics::M4.MM()`). If `NULL`, the fourth moment is not computed.

Details

The three quantities are defined as

$$m_2(w) = w' M_2 w, \quad m_3(w) = w' M_3 (w \otimes w), \quad m_4(w) = w' M_4 (w \otimes w \otimes w),$$

where $\otimes$ denotes the Kronecker product. These are the building blocks of the mean-variance-skewness-kurtosis (MVSK) framework of Jondeau, Poon and Rockinger (2007).

The function is a thin, exported wrapper around the package's internal helpers `.portm2`, `.portm3`, `.portm4` (which themselves replace the unexported `PerformanceAnalytics:::portm*` family). It exists so the textbook can demonstrate co-moment computation without reaching into non-exported symbols of the package via `:::`.

Value

A named list with components `m2`, `m3`, `m4` (`NULL` for any component whose corresponding co-moment matrix was not supplied).

References

Jondeau, E., Poon, S.-H., & Rockinger, M. (2007). *Financial Modeling Under Non-Gaussian Distributions*. Springer.

See Also

`f_ptf_max_U`, `f_efficient_frontier`

Examples

```
set.seed(1)
R  <- matrix(rnorm(200 * 4), 200, 4)
d  <- ncol(R)
M2 <- cov(R)
w  <- rep(1 / d, d)
# Variance of the equal-weighted portfolio:
f_portfolio_moments(w, M2 = M2)$m2

# Co-skewness / co-kurtosis matrices in d x d^2 / d x d^3 form (e.g. from
# PerformanceAnalytics::M3.MM() / M4.MM()) can be passed as M3, M4.
```

`f_ptf_max_U` *Maximize Truncated MVSK Utility with Box & Budget Constraints*

Description

Solves for portfolio weights that (approximately) maximize expected utility using a fourth-order moment expansion in mean–variance–skewness–kurtosis (MVSK). The problem is solved via SLSQP with the constraints $\sum_i w_i = 1$ and $0 \leq w_i \leq w_m ax$.

Usage

```
f_ptf_max_U(gamma, w_max, M1, M2, M3, M4)
```

Arguments

`gamma`	Non-negative numeric scalar risk-aversion parameter.
`w_max`	Numeric scalar in $(0, 1]$ giving the per-asset upper bound: $0 \leq w_i \leq w_m ax$.
`M1`	Numeric vector of expected returns (length d).
`M2`	Numeric $d \times d$ covariance matrix.
`M3`	Numeric $d \times d^2$ co-moment matrix, as returned by `M3.MM()` from the **PerformanceAnalytics** package.
`M4`	Numeric $d \times d^3$ co-moment matrix, as returned by `M4.MM()` from the **PerformanceAnalytics** package.

Details

The objective implemented is the negative of the truncated MVSK utility series, so minimizing it is equivalent to maximizing the utility:

$$\mathrm{EU}(w) \approx \mu_p - \frac{\gamma}{2}\sigma_p^2 + \frac{\gamma(\gamma+1)}{6}\, S_p - \frac{\gamma(\gamma+1)(\gamma+2)}{24}\, K_p,$$

where $\mu_p, \sigma_p^2, S_p, K_p$ are the portfolio's first four centralized moments and $\gamma \geq 0$ is a risk-aversion parameter.

The equality constraint is enforced via `eval_g_eq` and box constraints via `lb`/`ub`. Gradients are supplied analytically using internal helper functions '.derportm2', '.derportm3', and '.derportm4' defined in 'portfolio-moments.R'.

Value

A list with:

`w` Numeric vector of optimal portfolio weights (length d).

`EU` Scalar: value of the (approximate) expected utility at `w` (sign-corrected).

`status` Integer: the `nloptr` termination status code. Positive values indicate success; a warning is issued (and the possibly invalid solution returned) when the status is negative, i.e., when the optimizer failed to converge.

Implementation note

Portfolio moment calculations (`portm2/3/4` and their gradients) are performed by internal helpers in '`R/portfolio-moments.R`', vendored from the standard kronecker-product formulas (Jondeau et al., 2007). `M3` and `M4` must be co-moment matrices ($d \times d^2$ and $d \times d^3$), not higher-dimensional arrays.

References

Jondeau, E., Poon, S.-H., & Rockinger, M. (2007). *Financial Modeling Under Non-Gaussian Distributions*. Harvey, C. R., Liechty, J. C., Liechty, M. W., & Müller, P. (2010). Portfolio selection with higher moments.

See Also

`f_portfolio_moments`, `f_efficient_frontier`, `nloptr`

Examples

```
set.seed(1)
d <- 3
M1 <- c(0.06, 0.08, 0.07)
A  <- matrix(rnorm(d*d), d); M2 <- crossprod(A)/d
M3 <- matrix(0, d, d^2)
M4 <- matrix(0, d, d^3)
res <- f_ptf_max_U(gamma = 5, w_max = 0.8, M1, M2, M3, M4)
res$w; res$EU
```

`f_student_copula_pdf` *Multivariate Student-t Copula PDF*

Description

Computes the probability density function (PDF) of the multivariate Student-t copula at a specified point $u \in [0,1]^N$, given location vector μ, scatter matrix Σ, and degrees of freedom ν.

Usage

```
f_student_copula_pdf(u, mu, Sigma, nu, log = FALSE)
```

Arguments

`u`	Numeric vector of length N with entries in $(0,1)$: the evaluation point in the copula's domain.
`mu`	Numeric vector of length N, the location (mean) vector of the underlying multivariate Student-t distribution.
`Sigma`	Numeric $N \times N$ positive-definite scatter matrix. Typically a correlation matrix when defining a copula.
`nu`	Positive numeric scalar: degrees of freedom of the Student-t distribution ($\nu > 0$).
`log`	Logical; if `TRUE` the log-density is returned. Defaults to `FALSE`. In high dimension the density itself may overflow or underflow the double-precision range even though its logarithm is perfectly well behaved, so `log = TRUE` is the safe choice for likelihoods.

Details

The multivariate Student-t copula density is given by

$$c(u;\mu,\Sigma,\nu) = \frac{t_N\big(t_\nu^{-1}(u);\mu,\Sigma,\nu\big)}{\prod_{i=1}^N t_1\big(t_\nu^{-1}(u_i);\mu_i,\sigma_i^2,\nu\big)},$$

where t_N and t_1 denote the multivariate and univariate Student-t densities, respectively, and t_ν^{-1} is the quantile function of the univariate Student-t with ν degrees of freedom.

For $\nu \to \infty$, this copula converges to the Gaussian copula.

Value

A numeric scalar: the value of the Student-t copula density $c(u;\mu,\Sigma,\nu)$ at `u`, or its logarithm when `log = TRUE`. The value is returned as a plain numeric scalar (not a 1×1 matrix).

References

Demarta, S., & McNeil, A. J. (2005). The t Copula and Related Copulas. *International Statistical Review*, 73(1), 111–129. Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall. Nelsen, R. B. (2006). *An Introduction to Copulas* (2nd ed.). Springer.

See Also

f_normal_copula_pdf, f_clayton_copula_2d_pdf, f_gumbel_copula_2d_pdf

Examples

```
# Example: 2D t-copula density
mu <- c(0, 0)
Sigma <- matrix(c(1, 0.7, 0.7, 1), 2, 2)
f_student_copula_pdf(c(0.6, 0.8), mu, Sigma, nu = 5)

# Compare to Gaussian copula (nu large)
f_student_copula_pdf(c(0.6, 0.8), mu, Sigma, nu = 100)

# High dimension: use the log scale, the density itself overflows
d <- 300
f_student_copula_pdf(rep(0.99, d), rep(0, d), diag(d), nu = 5, log = TRUE)
```

---

f_tail_dependence *Empirical Tail Dependence and Exceedance Correlation*

Description

Estimates the empirical tail-dependence coefficient and the exceedance correlation between two series at a given quantile threshold α for either the lower or upper tail.

Usage

```
f_tail_dependence(x, y, alpha, side = c("lower", "upper"))
```

Arguments

x	Numeric vector of length n, first variable or return series.
y	Numeric vector of length n, second variable or return series.
alpha	Numeric scalar in $(0, 1)$, the tail probability level (e.g. 0.05 for the 5% tail).
side	Character string, either "lower" (default) or "upper", indicating which tail to analyse. "lower" conditions on $X < F_X^{-1}(\alpha)$; "upper" conditions on $X > F_X^{-1}(1-\alpha)$.

Details

The function classifies observations in the α-quantile tail of each margin and computes:

- the standard empirical tail-dependence coefficient: the proportion of joint tail events *conditional on* x being in its tail (lambda), and
- the sample correlation between the values of x and y for which both lie in their respective tails (excorr).

If the threshold selects no observation of `x` at all — which happens when `x` is constant or heavily tied — both components are `NA` and a warning is issued: an empty tail carries no information about tail dependence, and reporting zero there would describe a perfect match as independence.

If joint tail events are absent while the marginal tail is not empty, `lambda = 0` and `excorr = NA`. If fewer than two joint-tail observations are available, `excorr` is also `NA` (the sample correlation is undefined). When the joint-tail subset is degenerate (e.g., `x` and `y` are identical so that the subset has zero variance), `cor` returns `NA` with a warning.

Value

A list with components:

`lambda` Empirical tail-dependence coefficient. For `side = "lower"`:

$$\hat{\lambda}_L = \frac{\#\{X < q_\alpha^X \text{ and } Y < q_\alpha^Y\}}{\#\{X < q_\alpha^X\}}.$$

For `side = "upper"`:

$$\hat{\lambda}_U = \frac{\#\{X > q_{1-\alpha}^X \text{ and } Y > q_{1-\alpha}^Y\}}{\#\{X > q_{1-\alpha}^X\}}.$$

`excorr` Exceedance correlation between `x` and `y` in the joint tail (computed only if there are joint exceedances).

References

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall. Embrechts, P., McNeil, A. J., & Straumann, D. (2002). Correlation and dependence in risk management. *Risk Management: Value at Risk and Beyond*. Cambridge University Press.

See Also

`f_efficient_frontier`

Examples

```
set.seed(1)
x <- rnorm(1000)
y <- 0.7 * x + sqrt(1 - 0.7^2) * rnorm(1000)

f_tail_dependence(x, y, alpha = 0.05)
f_tail_dependence(x, y, alpha = 0.05, side = "upper")
```

GOLD *Gold Price (Daily, xts)*

Description

Daily World Gold Council gold price in USD per troy ounce, from 1970-01-01 to 2015-12-31.

Usage

```
data("GOLD")
```

Format

An `xts` object with 9691 daily observations and a single column `GOLD` containing USD prices per troy ounce.

Details

Originally distributed as `GOLD` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL (≥ 2) license as **qrmdata**.

Source

Federal Reserve Economic Data (FRED) via Quandl, downloaded on 2016-01-03 with `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("GOLD")
class(GOLD)        # "xts" "zoo"
dim(GOLD)
head(GOLD)
```

GoyalWelch *Goyal–Welch Predictive Variables (Monthly, U.S.)*

Description

Monthly time series of equity predictors and bond/credit variables commonly used in return predictability studies following Goyal & Welch (2008). Stored as an `xts` object indexed by month (class `zoo::yearmon`).

Usage

```
data("GoyalWelch")
```

Format

An `xts` object with monthly observations (Dec 1979 to Dec 2018 in this snapshot), indexed by `zoo::yearmon`, and 15 variables:

Index Broad U.S. equity price index level (e.g., S&P 500).

D12 Trailing 12-month cash dividends on the index (level).

E12 Trailing 12-month earnings on the index (level).

b/m Aggregate book-to-market ratio.

tbl 3-month Treasury bill rate, *annualized*, as a decimal (0.1204 = 12.04% per year). Observed monthly.

AAA Moody's AAA corporate bond yield, annualized decimal.

BAA Moody's BAA corporate bond yield, annualized decimal.

lty Long-term government bond yield, annualized decimal.

ntis Net equity expansion: shares issued less repurchases scaled by total equity (a supply measure).

Rfree Risk-free rate over the month, as a decimal. **This is exactly** `tbl / 12` — see the warning under Details.

infl Inflation (monthly change in price level).

ltr Long-term government bond total return over the month.

corpr Corporate bond total return over the month.

svar Stock market variance proxy (rolling sum of daily squared returns).

csp Cross-sectional premium of Polk, Thompson and Vuolteenaho (2006). Available from Dec 1979 to Dec 2002 only; the remaining 192 months are `NA`. It is *not* the corporate bond return spread, which is `corpr - ltr`.

Details

This object mirrors the variables used in Goyal & Welch (2008) and subsequent updates. From these levels you can form the standard ratios used in the literature, for example dp = log(D12/Index), ep = log(E12/Index), dfy = BAA - AAA, dfr = corpr - ltr and tms = lty - tbl.

Beware that the classical predictor list contains further exact redundancies of the same kind as tbl/Rfree: the payout ratio de = dp - ep and the term spread tms = lty - tbl are differences of other members. These predictors are used one at a time in univariate predictive regressions; entering them jointly with their components yields a singular design matrix.

Index is a price *level*, not a return. The equity premium is formed as (Index[t] + D12[t]/12) / Index[t-1] - 1 - Rfree[t].

Exact redundancy between tbl and Rfree

Rfree equals tbl / 12 to the last representable digit for every one of the 469 months: max(abs(GoyalWelch[,"tbl"]/12 -GoyalWelch[,"Rfree"])) is exactly zero. The two columns therefore carry the same information, and **at most one of them may enter a regression**. Passing both produces a rank-deficient design: lm() detects it and returns NA for the aliased coefficient, but penalized fitters such as glmnet::glmnet() do not — they split the effect arbitrarily between the two columns and report coefficients that can differ in sign from the identified solution. In predictive regressions Rfree normally belongs on the left-hand side, subtracted from the return to form the excess return.

Missing values

csp is missing for 192 of the 469 months (Jan 2003 onward). Every other column is complete. Selecting predictors positionally, for instance GoyalWelch[, 2:14], silently excludes csp; select by name instead so the exclusion is deliberate and visible.

Source

Compiled from the public Goyal–Welch data library (predictable stock returns) and standard fixed-income sources (e.g., FRED). See data-raw/GoyalWelch.R for the contract harness that pins the structural properties documented above.

References

Goyal, A., & Welch, I. (2008). A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. *Review of Financial Studies*, 21(4), 1455–1508.

Examples

```
data("GoyalWelch")
class(GoyalWelch)         # "xts" "zoo"
head(GoyalWelch)

# Rfree is exactly tbl / 12: never use both as predictors
max(abs(GoyalWelch[,"tbl"] / 12 - GoyalWelch[,"Rfree"]))

# The response: total return on the index, in excess of the risk-free rate
ix  <- GoyalWelch[,"Index"]
```

```
erp <- (ix + GoyalWelch[,"D12"] / 12) / xts::lag.xts(ix, 1) - 1 -
  GoyalWelch[,"Rfree"]
head(erp, 3)

# Construct common predictors:
dp  <- log(GoyalWelch[,"D12"]  / GoyalWelch[,"Index"])  # dividend-price
ep  <- log(GoyalWelch[,"E12"]  / GoyalWelch[,"Index"])  # earnings-price
dfy <- GoyalWelch[,"BAA"] - GoyalWelch[,"AAA"]          # default yield spread
dfr <- GoyalWelch[,"corpr"] - GoyalWelch[,"ltr"]        # default return spread
tms <- GoyalWelch[,"lty"] - GoyalWelch[,"tbl"]          # term spread
```

HSI *Hang Seng Index (Daily, xts)*

Description

Daily adjusted close prices of the Hang Seng stock index (ticker symbol `^HSI`), from its first date of availability on Yahoo Finance to 2015-12-31.

Usage

```
data("HSI")
```

Format

An `xts` object with 7214 daily observations and a single column `^HSI` containing adjusted close prices in index points. The time index spans from 1986-12-31 to 2015-12-31.

Details

Originally distributed as `HSI` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("HSI")
class(HSI)        # "xts" "zoo"
dim(HSI)
head(HSI)
```

`NIKKEI` *NIKKEI 225 Index (Daily, xts)*

Description

Daily adjusted close prices of the NIKKEI 225 stock index (ticker symbol `^N225`), from its first date of availability on Yahoo Finance to 2015-12-30.

Usage

```
data("NIKKEI")
```

Format

An `xts` object with 7880 daily observations and a single column `^N225` containing adjusted close prices in index points. The time index spans from 1984-01-04 to 2015-12-30.

Details

Originally distributed as `NIKKEI` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("NIKKEI")
class(NIKKEI)        # "xts" "zoo"
dim(NIKKEI)
head(NIKKEI)
```

SMI	*Swiss Market Index (Daily, xts)*

Description

Daily adjusted close prices of the Swiss Market Index (SMI) stock index (ticker symbol `^SSMI`), from its first date of availability on Yahoo Finance to 2015-12-30.

Usage

```
data("SMI")
```

Format

An `xts` object with 6350 daily observations and a single column `^SSMI` containing adjusted close prices in index points. The time index spans from 1990-11-09 to 2015-12-30.

Details

Originally distributed as `SMI` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**. Note: the **MSGARCH** package ships a different dataset also named `SMI` (SMI log-returns); to use that one, load it explicitly with `data("SMI", package = "MSGARCH")`.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("SMI")
class(SMI)       # "xts" "zoo"
dim(SMI)
head(SMI)
```

SP500 *S&P 500 Index (Daily, xts)*

Description

Daily adjusted close prices of the Standard & Poor's 500 stock index (ticker symbol `^GSPC`), from its first date of availability on Yahoo Finance to 2015-12-31.

Usage

```
data("SP500")
```

Format

An `xts` object with 16607 daily observations and a single column `^GSPC` containing adjusted close prices in index points. The time index spans from 1950-01-03 to 2015-12-31.

Details

Originally distributed as `SP500` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL (≥ 2) license as **qrmdata**.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("SP500")
class(SP500)        # "xts" "zoo"
dim(SP500)
head(SP500)
```

SP500_const *S&P 500 Constituents (Weekly, xts)*

Description

Weekly adjusted close prices for the 505 constituents of the S&P 500 index (membership as of the 2016-01-03 snapshot).

Usage

```
data("SP500_const")
```

Format

An `xts` object with 2818 weekly observations (the last available trading day of each week) and 505 columns, one per constituent (e.g., `AAPL`, `MSFT`, `XOM`). Missing values are common, especially in the early part of the sample, since a given name only has data once it became available. The time index spans from 1962-01-05 to 2015-12-31.

Unlike the `DJ_const`, `FTSE_const` and `EURSTX_const` panels (which are daily), this dataset is stored at **weekly** frequency to keep the package within the CRAN size limit. Weekly data is sufficient for the high-dimensional covariance and factor-model examples in the book, where a three-year window already has more constituents than observations.

Details

Originally distributed as `SP500_const` in the **qrmdata** package (Hofert, Hornik, & McNeil), thinned to weekly frequency and ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL ($\geq$ 2) license as **qrmdata**. The panel reflects index membership at the snapshot date and is therefore subject to survivorship bias.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("SP500_const")
class(SP500_const)        # "xts" "zoo"
dim(SP500_const)
head(colnames(SP500_const))
```

TermStructure *Daily Term Structure (1m–30y)*

Description

A compact term-structure object holding daily yield curves across 11 standard maturities from mid-2006 to late-2008. The object is an `xts` matrix of annualized yields (rows = dates, columns = maturities); the maturity grid in years is attached as the `"tau"` entry of `xts::xtsAttributes()`.

Usage

```
data("TermStructure")
```

Format

An `xts` object of annualized yields (percent, not decimals) with dimension 622×11. The index runs daily from `2006-05-12` to `2008-10-31`; columns are the 11 maturities: `X1mo`, `X3mo`, `X6mo`, `X1yr`, `X2yr`, `X3yr`, `X5yr`, `X7yr`, `X10yr`, `X20yr`, `X30yr`.

The maturity grid in years, `c(1/12, 1/4, 1/2, 1, 2, 3, 5, 7, 10, 20, 30)`, is stored as a user attribute and retrieved with `xts::xtsAttributes(TermStructure)$tau`.

Details

Dates run from `2006-05-12` to `2008-10-31` (about 622 business days in this snapshot). Yields are curve levels for each maturity on each date and can be used directly for fitting/parsing term-structure models (e.g., Nelson–Siegel or Svensson), computing spreads (term, slope, butterfly), or building zero curves.

Examples

```
data("TermStructure")
class(TermStructure)         # "xts" "zoo"
dts <- index(TermStructure)  # observation dates
colnames(TermStructure)

# Example: 10Y vs 2Y term spread (in percentage points)
sprd_10y2y <- TermStructure[, "X10yr"] - TermStructure[, "X2yr"]
head(sprd_10y2y)

# Maturity grid (years), stored as a user attribute
xts::xtsAttributes(TermStructure)$tau
```

VIX — *CBOE Volatility Index VIX (Daily, xts)*

Description

Daily close values of the Chicago Board Options Exchange (CBOE) volatility index VIX (ticker symbol `^VIX`), from its first date of availability on Yahoo Finance to 2015-12-31.

Usage

```
data("VIX")
```

Format

An `xts` object with 6553 daily observations and a single column `^VIX` containing the VIX level in percent (annualized implied volatility). The time index spans from 1990-01-02 to 2015-12-31.

Details

Originally distributed as `VIX` in the **qrmdata** package (Hofert, Hornik, & McNeil), ported into **smqf** so that the book's examples remain reproducible without an extra dependency. The data is redistributed here under the same GPL (≥ 2) license as **qrmdata**. The VIX is typically used as a market-based measure of volatility, expressed in percent.

Source

Yahoo Finance, downloaded on 2016-01-03 via `qrmtools::get_data()`.

References

Hofert, M., Hornik, K., & McNeil, A. J. *qrmdata: Data Sets for Quantitative Risk Management Practice*, https://CRAN.R-project.org/package=qrmdata.

Examples

```
data("VIX")
class(VIX)       # "xts" "zoo"
dim(VIX)
head(VIX)
```

Index