Package 'RcmdrPlugin.RiskDemo'

February 6, 2024

Type Package

Title R Commander Plug-in for Risk Demonstration

Version 3.2

Date 2024-02-05

Author Arto Luoma

Maintainer Arto Luoma <arto.luoma@wippies.com>

Description
R Commander plug-in to demonstrate various actuarial and financial risks. It includes valuation of bonds and stocks, portfolio optimization, classical ruin theory, demography and epidemic.

Depends R (>= 3.5.0)

Imports Rcmdr, demography, forecast, ftsa, ggplot2, dplyr, scales, zoo, data.table

Suggests tkrplot, rgl

License GPL-2

LazyData no

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2024-02-06 09:20:02 UTC

R topics documented:

`RcmdrPlugin.RiskDemo-package`

R Commander Plug-in for Risk Demonstration

Description

R Commander plug-in to demonstrate various actuarial and financial risks. It includes valuation of bonds and stocks, portfolio optimization, classical ruin theory, demography and epidemic.

Details

Package:	RcmdrPlugin.RiskDemo
Type:	Package
Version:	3.0
Date:	2021-04-03
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Arto Luoma

Maintainer: Arto Luoma <arto.luoma@wippies.com>

`bondCurve` *Drawing forward and yield curves*

Description

This function draws forward and yields curves, for AAA-rated central governement bonds and/or all central governement bonds.

Usage

```
bondCurve(date1, date2 = NULL, yield = TRUE, forward = TRUE,
  AAA = TRUE, all = TRUE, params)
```

Arguments

`date1`	The date for which the curves are drawn
`date2`	Optional second date for which the curves are drawn
`yield`	Is the yield curve shown (TRUE/FALSE)?
`forward`	Is the forward curve shown (TRUE/FALSE)?
`AAA`	Are the curves drawn for the AAA-rated bonds (TRUE/FALSE)?
`all`	Are the curves drawn for the bonds with all ratings (TRUE/FALSE)?
`params`	The data frame of curve parameters

Value

No value. Only a figure is produced.

Author(s)

Arto Luoma

References

https://bit.ly/2zfs0G8

Examples

```
data(params)
bondCurve(as.Date("2004-09-06"),params=params)
```

bondFigure *Bond price as a function of interest rate.*

Description

This function plots the bond price as a function of interest rate. It also shows, using dotted lines, the yield to maturity rate corresponding to the face value, and the flat price corresponding to the yield to maturity.

Usage

```
bondFigure(buyDate, matDate, rateCoupon, yieldToMat = NULL,
           bondPr = NULL, nPay)
```

Arguments

buyDate	the date when the coupon is bought (settlement date)
matDate	maturity date
rateCoupon	coupon rate (in decimals)
yieldToMat	yield to maturity (in decimals)
bondPr	the flat price of the bond
nPay	number of coupon payments per year

Details

either yieldToMat or bondPr should be given as input.

Value

This function only plots a figure.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Bodie, Kane, and Marcus (2014) *Investments, 10th Global Edition*, McGraw-Hill Education, (see Section 14.2 Bond Pricing).

See Also

bondPrice, solveYield

Examples

```
bondFigure("2012-7-31","2018-7-31",rateCoupon=0.0225,yieldToMat=0.0079,
           nPay=2)
bondFigure("2012-7-31","2018-7-31",rateCoupon=0.0225,bondPr=90,nPay=2)
```

`bondPrice` *Computing bond prices*

Description

This function computes the bond price, given the yield to maturity.

Usage

```
bondPrice(buyDate, matDate, rateCoupon, yieldToMat, nPay)
```

Arguments

`buyDate`	the date at which the bond is bought (settlement date).
`matDate`	maturity date
`rateCoupon`	annual coupon date
`yieldToMat`	yield to maturity
`nPay`	number of coupon payments per day

Details

All the rates are given in decimals.

Value

A list with the following components:

`yieldToMaturity`	yield to maturity
`flatPrice`	flat price
`daysSinceLastCoupon`	days since previous coupon payment
`daysInCouponPeriod`	days in a coupon period
`accruedInterest`	accrued interest since last coupon payment
`invoicePrice`	invoice price (= flat price + accrued interest)

Note

With Excel functions PRICE, DATE, COUPDAYBS and COUPDAYS you can do the same.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Bodie, Kane, and Marcus (2014) *Investments, 10th Global Edition*, McGraw-Hill Education, (see Bond Pricing between Coupon Dates in Section 14.2).

See Also

solveYield

Examples

```
bondPrice("2012-7-31","2018-7-31",0.0225,0.0079,2)
bondPrice("2012-7-31","2018-7-31",0.0225,0.0079,4)
bondPrice("2012-7-31","2030-5-15",0.0625,0.02117,2)
```

---

computeRuin *Ruin probability computation with infinite time horizon*

Description

This function uses classical ruin theory to compute either ruin probability, safety loading or initial capital, given two of them. The time horizon is infinite. Gamma distribution is used to model claim sizes.

Usage

```
computeRuin(U0 = NULL, theta = NULL, eps = NULL, alpha, beta)
```

Arguments

U0	initial capital
theta	safety loading
eps	ruin probability
alpha	shape parameter of gamma distribution
beta	rate parameter of gamma distribution

Value

The value is a list with the following components:

LundbergExp	Lundberg's exponent R
initialCapital	initial capital
safetyLoading	safety loading
ruinProb	ruin probability

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Gray and Pitts (2012) *Risk Modelling in General Insurance: From Principles to Practice*, Cambridge University Press.

See Also

computeRuinFinite, solveLund

Examples

```
computeRuin(U0=1000,theta=0.01,alpha=1,beta=0.1)
computeRuin(eps=0.005,theta=0.01,alpha=1,beta=0.1)
computeRuin(U0=5399.24,eps=0.005,alpha=1,beta=0.1)
```

---

computeRuinFinite *Ruin probability computation with finite time horizon*

Description

This function uses classical ruin theory to compute either ruin probability, safety loading or initial capital, given two of them. The time horizon is finite. Gamma distribution is used to model claim sizes.

Usage

```
computeRuinFinite(T0, U0 = NULL, theta = NULL, eps = NULL, lambda,
                  alpha, beta)
```

Arguments

T0	time horizon (in years)
U0	initial capital
theta	safety loading
eps	ruin probability
lambda	claim intensity (mean number of claims per year)
alpha	shape parameter of gamma distribution
beta	rate parameter of gamma distribution

Value

The value is a list with the following components:

LundbergExp	Lundberg's exponent R
initialCapital	initial capital
safetyLoading	safety loading
ruinProb	ruin probability

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

computeRuin, solveLund

Examples

```
computeRuinFinite(T0=100,U0=1000,theta=0.01,lambda=100,alpha=1,beta=0.1)
computeRuinFinite(T0=1,eps=0.005,theta=0.001,lambda=100,alpha=1,beta=0.1)
computeRuinFinite(T0=500,U0=5347,eps=0.005,lambda=100,alpha=1,beta=0.1)
```

---

countries.mort *Mortality data*

Description

Mortality data for 10 countries (period death rates and exposures) retrieved from Human Mortality Database. The data are rounded to three significant digits and include the Nordic countries, China, U.S., Russia, Japan and Germany.

Usage

```
data("countries.mort")
```

Format

List of objects of class demogdata.

Source

HMD. Human Mortality Database. Max Planck Institute for Demographic Research (Germany), University of California, Berkeley (USA), and French Institute for Demographic Studies (France). Available at www.mortality.org. (Data downloaded Nov 13, 2023.)

Examples

```
data(countries.mort)
plot(countries.mort[[1]])
```

`covidSmooth` *Kalman smoothing of the covid model*

Description

This function does Kalman smoothing for the simple model that is used to predict new COVID-19 cases.

Usage

```
covidSmooth(par, y)
```

Arguments

`par`	Logarithms of the variance parameters of drift, seasonal component, and error term
`y`	Univariate numeric time series of new COVID-19 cases

Details

See loglikCovid.

Value

`Xif`	Matrix of filtered values, where the state vectors are given as rows
`Xis`	Matrix of smoothed values, where the state vectors are given as rows
`Pmat`	Array of state uncertainty matrices, evaluated at time *t-1*. The first array index is for time.
`Pfmat`	Array of state uncertainty matrices, evaluated at time *t*. The first array index is for time.
`Psmat`	Array of state uncertainty matrices, evaluated at time *n*, where *n* is the number of observations. The first array index is for time.

Author(s)

Arto Luoma

See Also

`loglikCovid`

Examples

```
#Preparing a time series
library(zoo)
data(dataCovidFin)
timeindex <- dataCovidFin[dataCovidFin$Alue=="Kaikki Alueet","Aika"]
series <- dataCovidFin[dataCovidFin$Alue=="Kaikki Alueet","val"]
series <- window(zoo(series,order.by=timeindex),start="2020-03-01",
                 end="2021-03-01")
#Fitting a state space model and smoothing the components
p0 <- c(-9,-7,-3.3)
fit <- nlm(loglikCovid,p=p0,y=series)
out <- covidSmooth(fit$estimate,y=series)

#Plotting the filtered and smoothed components
smoothed <- zoo(out$Xis[,1:3],order.by=time(series))
filtered <- zoo(out$Xif[,1:3],order.by=time(series))
colnames(smoothed) <- colnames(filtered) <- c("Level","Drift","Seasonal")
plot(filtered,xlab="Time",main="Filtered components of the time series")
plot(smoothed,xlab="Time",main="Smoothed components of the time series")

#Plotting the original time series, and the filtered and smoothed local level
#series after transforming them to original scale
plot(series,xlab="Time",ylab="Time series")
lines(exp(filtered[,1])-2,col=3)
lines(exp(smoothed[,1])-2,col=2)
legend("topleft",c("original","filtered","smoothed"),col=c(1,3,2),lty=1)
```

---

dataCovid *COVID-19 statistics*

Description

This data set consists of several statistics about the COVID-19 pandemic in 45 countries.

Usage

```
data("dataCovid")
```

Format

A data frame with 18400 observations on the following 27 variables.

`location` a character vector

`date` a Date

`new_cases` a numeric vector

`new_cases_per_million` a numeric vector

`new_cases_smoothed_per_million` a numeric vector

`new_cases_smoothed` a numeric vector

`new_deaths_per_million` a numeric vector

`new_deaths` a numeric vector

`new_deaths_smoothed_per_million` a numeric vector

`new_deaths_smoothed` a numeric vector

`total_deaths_per_million` a numeric vector

`total_deaths` a numeric vector

`total_cases` a numeric vector

`total_cases_per_million` a numeric vector

`hosp_patients` a numeric vector

`hosp_patients_per_million` a numeric vector

`icu_patients_per_million` a numeric vector

`icu_patients` a numeric vector

`reproduction_rate` a numeric vector

`new_tests` a numeric vector

`new_tests_per_thousand` a numeric vector

`tests_per_case` a numeric vector

`positive_rate` a numeric vector

`new_tests_smoothed` a numeric vector

`new_tests_smoothed_per_thousand` a numeric vector

`total_tests` a numeric vector

`total_tests_per_thousand` a numeric vector

Details

This is a subset of the complete data set available online, downloaded on March 31, 2021.

Source

https://covid.ourworldindata.org/data/owid-covid-data.csv

Examples

```
library(zoo)
data(dataCovid)
casesFin <- subset(dataCovid,subset=location=="Finland", select=c(date,new_cases))
plot(zoo(casesFin$new_cases,order.by=casesFin$date),ylab="New COVID-19 cases in Finland",
xlab="")
```

`dataCovidFin` *Confirmed COVID-19 cases in Finland*

Description

This data set provides the confirmed COVID-19 cases in 21 Finnish hospital districts, in addition to the total number.

Usage

```
data("dataCovidFin")
```

Format

A data frame with 16082 observations on the following 3 variables.

`Aika` Date

`Alue` character vector: hospital district

`val` numeric vector: number of new confirmed cases

Details

The data were downloaded on March 31, 2021, via THL's open data API.

Source

https://bit.ly/2PO1DnS

References

https://bit.ly/3ryfwE4

Examples

```
library(zoo)
data(dataCovidFin)
casesFin <- subset(dataCovidFin, subset = Alue=="Kaikki Alueet")
plot(zoo(casesFin$val,order.by=casesFin$Aika),ylab="New COVID-19 cases in Finland",xlab="")
```

drawBars *Plotting epidemic statistics*

Description

This function plots several epidemic statistics for selected countries.

Usage

```
drawBars(data, countries, start = "2020-06-01", end = "last", measure = "new_cases",
    atop = TRUE, perMillion = FALSE, drawMean = TRUE, bars = TRUE)
```

Arguments

data	data frame similar to (or including the same columns as) dataCovid
countries	vector of characters srings indicating the countries for which the selected statistic is plotted
start	beginning date of the time window for which the statistic is plotted
end	ending date of the time window for which the statistic is plotted
measure	statistic to be plotted
atop	logical indicating if the bars of different countries are plotted on top of one another
perMillion	logical indicating if the statistic is proportioned to a population of million
drawMean	logical indicating if a smoothed curve is drawn
bars	logical indicating if bars are plotted

Value

No value.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

drawBarsFin, dataCovid

Examples

```
data(dataCovid)
drawBars(data=dataCovid, countries=c('Finland','France'),start='2020-6-1',
         measure='new_cases',perMillion=TRUE)
```

drawBarsFin *Plotting epidemic statistics with Finnish data*

Description

This function plots the new cases or total cases of an epidemic for selected regions in Finland.

Usage

```
drawBarsFin(data, pop, regions, start = "2020-06-01", end = "last",
    measure = "new_cases", atop = TRUE, perMillion = FALSE, drawMean = TRUE,
    bars = TRUE)
```

Arguments

data	data frame including columns Aika (character string indicating the date), Alue (character string indicating the region) and val (numeric indicating the number of new cases)
pop	data frame including columns Alue (character string indicating the region) and val (integer indicating the population)
regions	vector of characters strings indicating the regions for which the selected statistic is plotted
start	beginning date of the time window for which the curve is plotted
end	ending date of the time window for which the curve is plotted
measure	statistic to be plotted
atop	logical indicating if the bars of different regions are plotted on top of one another
perMillion	logical indicating if the statistic is proportioned to a population of million
drawMean	logical indicating if a smoothed curve (rolling mean of 7 observations) is plotted
bars	logical indicating if bars are plotted

Value

No value.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

drawBars, dataCovidFin

Examples

```
data(dataCovidFin)
data(popRegionsFin)
drawBarsFin(dataCovidFin,popRegionsFin,regions=popRegionsFin$Alue[1:7])
```

drawFigure *Efficient frontier and return distribution figures*

Description

Plots the efficient frontiers of risky investments and all investments. The optimum points corresponding to the risk aversion coefficient are indicated by dots. Further, the function plots a predictive return distribution figure.

Usage

```
drawFigure(symbol, yield, vol, beta, r = 1,
  total = 1, indexVol = 20, nStocks = 7, balanceInt = 12, A = 10,
  riskfree = FALSE, bor = FALSE)
```

Arguments

symbol	character vector of the symbols of the risky investments
yield	vector of yields (%)
vol	vector of volatilities (%)
beta	vector of betas (%)
r	risk-free interest rate (%)
total	total investment (for example in euros)
indexVol	volatility of market portfolio (%)
nStocks	number of risky investments in the portfolio
balanceInt	balancing interval of the portfolio in months
A	risk aversion coefficient (see details)
riskfree	is risk-free investment included in the portfolio (logical)
bor	is borrowing (negative risk-free investment) allowed (logical)

Details

The function uses the single-index model and Markovitz portfolio optimization model to find the optimum risky portfolio. The returns are assumed to be log-normally distributed. The maximized function is mu - 0.5*A*var where mu is expected return, A is risk aversion coefficient, and var is return variance.

Value

portfolio	allocation of the total investment (in euros)
returnExpectation	expected portfolio return
returnDeviation	standard deviation of the portfolio

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Bodie, Kane, and Marcus (2014) *Investments, 10th Global Edition*, McGraw-Hill Education, (see Section 7.4 The Markowitz Portfolio Optimization Model and Section 8.2 The Single-Index Model).

See Also

portfOptim

Examples

```
data(stockData, package="RcmdrPlugin.RiskDemo")
with(stockData,drawFigure(symbol=rownames(stockData),yield=divYield,
  vol=vol,beta=beta,r=1,total=100,indexVol=10,
  nStocks=5,balanceInt=12,A=10,riskfree=TRUE,bor=FALSE))
```

---

drawIncidence *Plotting incidence curves of an epidemic*

Description

This function plots incidence curves of an epidemic for selected countries. The incidences are new cases per 100 000 inhabitants within one or two weeks.

Usage

```
drawIncidence(data, countries, start = "2020-06-01", end = "last", weeks = 2,
    log = TRUE)
```

Arguments

data	data frame including columns location (character string indicating the country), date (character string) and new_cases_per_million (numeric)
countries	vector of characters srings indicating the countries for which the curves are plotted
start	beginning date of the time window for which the curve is plotted
end	ending date of the time window for which the curve is plotted
weeks	Integer telling how many weeks' observations are used to calculate the incidence. Usually 1 or 2.
log	logical indicating if a log scale is used in the plot

Value

No value

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

drawIncidenceFin, dataCovid

Examples

```
data(dataCovid)
Europe <- c("Germany","France","United Kingdom","Italy","Spain","Poland","Romania",
    "Netherlands","Belgium","Greece")
drawIncidence(dataCovid,countries=Europe)
```

---

drawIncidenceFin *Plotting incidence curves of an epidemic with Finnish data*

Description

This function plots incidence curves of an epidemic for selected regions of Finland. The incidences are new cases per 100 000 inhabitants within one or two weeks.

Usage

```
drawIncidenceFin(data, pop, regions, start = "2020-06-01", end = "last", weeks = 2,
    includeAllRegions = TRUE, log = TRUE)
```

Arguments

data	data frame including columns Aika (character string indicating the date), Alue (character string indicating the region) and val (numeric indicating the number of new cases)
pop	data frame including columns Alue (character string indicating the region) and val (integer indicating the population)
regions	vector of characters srings indicating the regions for which the curves are plotted
start	beginning date of the time window for which the curve is plotted
end	ending date of the time window for which the curve is plotted
weeks	Integer telling how many weeks' observations are used to calculate the incidence. Usually 1 or 2.
includeAllRegions	logical indicating if a curve for total incidence is included
log	logical indicating if a log scale is used in the plot

Value

No value

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

drawIncidence, dataCovidFin

Examples

```
data(dataCovidFin)
data(popRegionsFin)
drawIncidenceFin(data = dataCovidFin, pop = popRegionsFin,
    regions = popRegionsFin$Alue[1:5], start = "2020-06-01", end="last", weeks=2,
    includeAllRegions = TRUE)
```

---

drawPositiveRate *Plotting the positive rate of COVID-19 tests or the tests per case*

Description

This function plots a time series of either the positive rate of COVID-19 tests or the number of tests per case.

Usage

```
drawPositiveRate(data, countries, start = "2020-06-01", end = "last",
    measure = "positive_rate", curve = TRUE, bars = FALSE, log = FALSE)
```

Arguments

data	data frame including columns location (character string indicating the country), date (character string) and tests_per_case, positive_rate (numeric)
countries	vector of characters srings indicating the countries for which the selected statistic is plotted
start	beginning date of the time window for which the time series are plotted
end	ending date of the time window for which the time series are plotted
measure	statistic for which the time series are plotted
curve	logical indicating if smoothed curves are drawn
bars	logical indicating if bars are plotted
log	logical indicating if a log scale is used in the plot

Value

No value.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

dataCovid, drawTests

Examples

```
data(dataCovid)
drawPositiveRate(dataCovid,countries=c("Finland","France"))
```

---

drawRuin *Plotting simulations of a surplus process*

Description

This function plots simulation paths of a surpluss process. The claims are assumed to arrive according to a Poisson process and the claim sizes are assumed to be gamma distributed.

Usage

```
drawRuin(nsim = 10, Tup = 10, U0 = 1000, theta = 0.01,
         lambda = 100, alpha = 1, beta = 0.1)
```

Arguments

nsim	number of simulations
Tup	maximum value in the time axis
U0	initial capital
theta	risk loading
lambda	intensity of claim process (mean number of claims per year)
alpha	shape parameter of gamma distribution
beta	rate parameter of gamma distribution

Value

No value; only a figure is plotted.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Kaas, Goovaerts, Dhaene, Denuit (2008) *Modern actuarial risk theory using R, 2nd ed.*, Springer.

See Also

computeRuinFinite,

Examples

```
computeRuinFinite(T0=10,U0=1000,eps=0.05,lambda=100,alpha=1,beta=0.1)
drawRuin(nsim=10,Tup=10,U0=1000,theta=0.0125,lambda=100,alpha=1,beta=0.1)
```

---

drawTests *Plotting time series related to COVID-19 testing*

Description

This function plots time series of new and total COVID-19 tests, possibly in proportion to population.

Usage

```
drawTests(data, countries, start = "2020-06-01", end = "last", measure = "new_tests",
    atop = TRUE, perThousand = FALSE, drawMean = TRUE, bars = TRUE, log = FALSE)
```

Arguments

data	data frame similar to (or including the same columns as) dataCovid
countries	vector of characters strings indicating the countries for which the time series are plotted
start	beginning date of the time window for which the time series are plotted
end	ending date of the time window for which the time series are plotted
measure	statistic for which the time series are plotted
atop	logical indicating if the bars of different countries are plotted on top of one another
perThousand	logical indicating if the statistic is proportioned to a population of thousand
drawMean	logical indicating if a smoothed curve is drawn
bars	logical indicating if bars are plotted
log	logical indicating if a log scale is used in the plot

Value

No value.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

dataCovid, drawPositiveRate

Examples

```
data(dataCovid)
drawTests(dataCovid,countries=c("Finland","France"),perThousand=TRUE)
```

---

fin *Mortality data for Finland*

Description

Mortality data for Finland Series: female male total Years: 1878 - 2015 Ages: 0 - 110

Usage

```
data("fin")
```

Format

object of class demogdata

Details

This is part of the countries.mort data (countries.mort[[11]]).

Source

HMD. Human Mortality Database. Max Planck Institute for Demographic Research (Germany), University of California, Berkeley (USA), and French Institute for Demographic Studies (France). Available at www.mortality.org. (Data downloaded Nov 13, 2023.)

Examples

```
data(fin)
print(fin)
plot(fin)
```

fin.fcast *Finnish mortality forecast*

Description

Finnish mortality forecast 50 years ahead (2023-2072) for 0 - 100 years old. The forecast is based on an estimated Lee-Carter model. The kt coefficients were forecast using a random walk with drift. Fitted rates were used as the starting value.

Usage

```
data("fin.fcast")
```

Format

An object of class "fmforecast"; for details, see documentation of package "demography".

Details

The forecast was produced using function "forecast.lca" of package "demography".

Examples

```
data(fin.fcast)
print(fin.fcast)
plot(fin.fcast)
```

fin.lca *Lee-Carter model fit for Finnish data*

Description

Lee-Carter model fit obtained by function "lca" of package "demography". The fit is based on Finnish mortality data for ages from 0 to 100 and years from 1950 to 2022.

Usage

```
data("fin.lca")
```

Format

object of class "lca"

Details

Both sexes were included in the input mortality data.

Examples

```
data(fin.lca)
plot(fin.lca)
```

`loglikCovid` *Computing the log-likelihood of the covid model*

Description

This function computes -2 times the log-likelihood of the simple model that is used to predict new COVID-19 cases and to estimate the effective reproduction number.

Usage

```
loglikCovid(y, par, it = TRUE)
```

Arguments

`y`	Univariate numeric time series of new COVID-19 cases
`par`	Logarithms of the variance parameters of drift, seasonal component, and error term
`it`	A logical value indicating if only the log-likelihood is returned.

Details

Some multiplicative and additive constants are omitted when the negative log-likelihood is computed. Before computing the log-likelihood, the transformation *y=log(x+a)*, where *a=2*, is applied to the time series. The model is a simple local linear model with local level, drift and seasonal component. The variance parameters of the level and seasonal component are estimated while the variance of the level component is computed as *max(exp(xi[1]) - a, 0.1)/exp(xi[1])^2*, where *xi[1]* is the current estimate of the level. This is based on the assumption that the number of new cases is approximately Poisson distributed, so that the variance equals the level. The *max* operation is taken in order to prevent the exression from being negative. In order to facilitate estimation, a penalty term is added which corresponds to a prior of *N(-9,1)* for the logarithm of the drift variance.

Value

`loglik`	-2 times the penalized log likelihood apart from some additive constants
`ll`	Vector of the increments of the log-likelihood corresponding to individual observations
`Xi`	Matrix of one-step predictions of the state vector. The vectors at different time points are given as rows.
`Xif`	Matrix of filtered values, where the state vectors are given as rows
`Pfmat`	Array of state uncertainty matrices, evaluated at time *t*. The first array index is for time.
`Q`	Covariance matrix of the error vector of the state equation

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Hamilton (1994) *Time Series Analysis*, Princeton University Press, (see Chapter 13 The Kalman Filter).

See Also

covidSmooth

Examples

```
#See examples for covidSmooth.
```

`params` *Yield curve parameter data*

Description

Yield curve parameters from the European Central Bank (ECB), downloaded on Nov 4, 2023

Usage

```
data("params")
```

Format

A data frame with 4902 observations on the following 13 variables.

`date` a Date

`b0` a numeric vector

`b1` a numeric vector

`b2` a numeric vector

`b3` a numeric vector

`t1` a numeric vector

`t2` a numeric vector

`c0` a numeric vector

`c1` a numeric vector

`c2` a numeric vector

`c3` a numeric vector

`d1` a numeric vector

`d2` a numeric vector

Details

The parameters b0 to b3 are the beta-parameters, and t1 and t2 the tau-parameters for AAA-rated government bonds. The parameters c0 to c3 are the beta-parameters, and d1 and d2 the tau-parameters for all government bonds.

Source

https://bit.ly/2zfs0G8

Examples

```
data(params)
bondCurve(as.Date("2004-09-06"),params=params)
```

`plotForecast` *Forecasting new covid cases*

Description

This function forecasts the numbers of new covid cases using a simple linear state space model.

Usage

```
plotForecast(data, region, start = NULL, end = NULL, np = 30, predInt = 0.95,
    log = TRUE)
```

Arguments

`data`	data frame including columns `Aika` (character string indicating the date), `Alue` (character string indicating the region) and `val` (numeric indicating the number of new cases)
`region`	characters string indicating the region for which the forecast is made
`start`	beginning date of the observations used in the estimation of the forecasting model
`end`	ending date of the observations used in the estimation of the forecasting model
`np`	integer indicating the forecasting horizon in days
`predInt`	decimal indicating the probability of the forecasting interval
`log`	logical indicating if a log scale is used in the plot

Value

No value.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

plotR, dataCovidFin

Examples

```
data(dataCovidFin)
plotForecast(data=dataCovidFin, region='All regions', start="2020-09-01")
```

---

plotR *Plotting the effective reproduction number (R)*

Description

This function plots a time series of the effective reproduction number R and its confidence interval.

Usage

```
plotR(data, region, start = NULL, end = NULL, confInt = 0.95)
```

Arguments

data	data frame including columns Aika (character string indicating the date), Alue (character string indicating the region) and val (numeric indicating the number of new cases)
region	characters string indicating the region for which the R series is computed
start	beginning date of the time window for which the R is computed
end	ending date of the time window for which the R is computed
confInt	decimal between 0 and 1, indicating the level of the confidence interval of R

Value

No value

Author(s)

Arto Luoma <arto.luoma@wippies.com>

See Also

plotForecast, dataCovidFin

Examples

```
data(dataCovidFin)
plotR(data=dataCovidFin, region='All regions')
```

pop.pred *Population forecasting*

Description

Population forecasting using mortality forecast and simple time series forecast for age 0 population

Usage

```
pop.pred(mort, mort.fcast)
```

Arguments

mort	mortality data of class 'demogdata'
mort.fcast	mortality forecast of class 'fmforecast'

Details

ARIMA(0,2,2)-model is used to forecast age 0 populaton.

Value

population forecast of class 'demogdata'

Author(s)

Arto Luoma <arto.luoma@wippies.com>

Examples

```
data(fin)
data(fin.fcast)
fin.pcast <- pop.pred(fin,fin.fcast)
plot(fin,plot.type="functions",series="total",transform=FALSE,
     datatype="pop",ages=c(0:100), years=c(1990+0:5*10), xlab="Age")
lines(fin.pcast,plot.type="functions",series="total",transform=FALSE,
      datatype="pop",ages=c(0:100), years=c(1990+0:5*10), lty=2)
```

popRegionsFin *Population data on Finnish hospital districts*

Description

This data set provides the populations of the 21 hospital districts, in addition to the total Finnish population.

Usage

```
data("popRegionsFin")
```

Format

A data frame with 22 observations on the following 2 variables.

`Alue` character vector: hospital district

`val` numeric vector: population

Details

The data were downloaded on March 31, 2021, via THL's open data API.

Source

https://bit.ly/39uZy7C

References

https://bit.ly/3ryfwE4

Examples

```
data(popRegionsFin)
print(popRegionsFin)
```

`portfOptim` *Portfolio optimization for an index model*

Description

Finds an optimal portfolio for long-term investments and plots a return distribution.

Usage

```
portfOptim(i, symbol, yield, vol, beta,
  indexVol = 0.2, nStocks = 7, total = 1, balanceInt = 1,
  C = 0.05, riskProportion = 1, riskfreeRate = 0, sim = FALSE)
```

Arguments

`i`	vector of the indices of the included risky investments
`symbol`	character vector of the symbols of the risky investments
`yield`	vector of expected yields (in euros)
`vol`	vector of volatilities
`beta`	vector of betas
`indexVol`	portfolio index volatility
`nStocks`	number of stocks in the portfolio
`total`	total sum invested (in euros)
`balanceInt`	balancing interval of the portfolio (in years)
`C`	expected portfolio return (in euros)
`riskProportion`	proportion of risky investments
`riskfreeRate`	risk-free interest rate
`sim`	is the return distribution simulated and plotted (logical value)?

Details

The arguments vol, beta, indexVol, riskProportion and riskfreeRate are given in decimals. The portfolio is optimized by minimizing the variance of the portfolio yield for a given expected yield. The returns are assumed to be log-normally distributed. The covariance matrix is computed using the single index model and the properties of the log-normal distribution.

Value

`portfolio`	numeric vector of allocations to each stock (in euros)
`returnExpectation`	expected value of the return distribution (in euros)
`returnDeviation`	standard deviation of the return distribution (in euros)
`VaR`	0.5%,1%,5%,10% and 50% percentiles of the return distribution (in euros)

Note

This function is usually called by drawFigure.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Bodie, Kane, and Marcus (2014) *Investments, 10th Global Edition*, McGraw-Hill Education, (see Section 7.4 The Markowitz Portfolio Optimization Model and Section 8.2 The Single-Index Model).

See Also

drawFigure

Examples

```
data(stockData, package="RcmdrPlugin.RiskDemo")
with(stockData,portfOptim(i=1:5,symbol=rownames(stockData),
  yield=divYield/100,vol=vol/100,beta=beta/100,total=100, sim=TRUE))
```

---

returns *Computing expected returns and their covariance matrix*

Description

Computing expected returns and their covariance matrix when the returns are lognormal.

Usage

```
returns(volvec, indexvol, beta)
```

Arguments

volvec	vector of volatilities
indexvol	volatility of the portfolio index
beta	vector of betas

Details

The arguments are given in decimals. The single index model is used to compute the covariance matrix of a multivariate normal distribution. The mean vector is assumed to be zero. The properties of the log-normal distribution are then used to compute the mean vector and covariance matrix of the corresponding multivariate log-normal distribution.

Value

mean	vector of expected returns
cov	covariance matrix of returns

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Bodie, Kane, and Marcus (2014) *Investments, 10th Global Edition*, McGraw-Hill Education, (see Section 8.2 The Single-Index Model).

Examples

```
returns(volvec=c(0.1,0.2,0.3),indexvol=0.2, beta=c(0.5,-0.1,1.1))
```

---

solveLund *Solving Lund's exponent*

Description

This function solves Lund's exponent or adjustment coefficient. The claim sizes are assumed to be gamma distributed.

Usage

```
solveLund(alpha, beta, theta)
```

Arguments

alpha	shape parameter of gamma distribution
beta	rate parameter of gamma distribution
theta	safety loading

Value

Lundberg's exponent (or adjustment coefficient)

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Gray and Pitts (2012) *Risk Modelling in General Insurance: From Principles to Practice*, Cambridge University Press.

See Also

computeRuin, computeRuinFinite

Examples

```
solveLund(1,1,0.1)
```

`solveYield` *Computing bond yields*

Description

This function computes the yield to maturity, given the (flat) bond price.

Usage

```
solveYield(buyDate, matDate, rateCoupon, bondPr, nPay)
```

Arguments

`buyDate`	settlement date (the date when the bond is bought)
`matDate`	maturity date
`rateCoupon`	annual coupon rate
`bondPr`	bond price. The flat price without accrued interest.
`nPay`	number of payments per year

Details

all the rates are given in decimals

Value

A list with the following components:

`yieldToMaturity`	yield to maturity
`flatPrice`	flat price
`daysSinceLastCoupon`	days since previous coupon payment
`daysInCouponPeriod`	days in a coupon period
`accruedInterest`	accrued interest since last coupon payment
`invoicePrice`	invoice price (= flat price + accrued interest)

Note

With Excel function YIELD you can do the same.

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Bodie, Kane, and Marcus (2014) *Investments, 10th Global Edition*, McGraw-Hill Education, (see Bond Pricing between Coupon Dates in Section 14.2).

See Also

bondPrice

Examples

```
solveYield("2012-7-31","2018-7-31",0.0225,100,2)
```

---

stock.price *Computing stock prices*

Description

This function computes the intrinsic stock price using the constant growth dividend discount model.

Usage

```
stock.price(dividend, k = NULL, g = NULL, ROE = NULL, b = NULL,
  riskFree = NULL, marketPremium = NULL, beta = NULL)
```

Arguments

dividend	expected dividend(s) for the next year(s) (in euros), separated by commas
k	required rate of return
g	growth rate of dividends
ROE	return on investment
b	plowback ratio
riskFree	riskfree rate
marketPremium	market risk premium
beta	beta

Details

All the above rates are given in percentages (except the dividends). One should provide either k or the following three: riskFree, marketPremium, beta. Further, one should provide either g or the following two: ROE and b. In the output, k and g are given in decimals.

Value

`dividend`	expected dividend(s) for the next year(s) (in euros)
`k`	required rate of return
`g`	growth rate of dividends
`PVGO`	present value of growths opportunities
`stockPrice`	intrinsic stock price

Author(s)

Arto Luoma <arto.luoma@wippies.com>

References

Bodie, Kane, and Marcus (2014) *Investments, 10th Global Edition*, McGraw-Hill Education, (see Dividend Discount Models in Section 18.3).

Examples

```
stock.price(dividend=c(1),k=12,g=10)
stock.price(dividend=c(1),ROE=50,b=20,riskFree=5,marketPremium=8,
            beta=90)
```

`stockData` *Stock data*

Description

Stock data on large companies in Helsinki Stock Exchange, downloaded from Kauppalehti web page (www.kauppalehti.fi), on May 13, 2017

Usage

```
data("stockData")
```

Format

A data frame with 35 observations on the following 7 variables.

`names` name of the firm

`abbrs` abbreviation of the firm

`quote` closing quote

`vol` volatility (%)

`beta` beta (%)

`div` dividend (eur/stock)

`divYield` dividend yield (%)

Source

www.kauppalehti.fi

Examples

```
data(stockData)
plot(stockData[,-(1:2)])
```

Index