

Package: R4GoodPersonalFinances (via r-universe)

December 17, 2024

Title Make Better Financial Decisions

Version 0.0.0.9025

Description Make informed, data-driven decisions for your personal or household finances. Use tools and methods that are selected carefully to align with academic consensus, bridging the gap between theoretical knowledge and practical application. They assist you in finding optimal asset allocation, preparing for retirement or financial independence, calculating optimal spending, and more.

License MIT + file LICENSE

URL <https://www.r4good.academy/>,
<https://r4goodacademy.github.io/R4GoodPersonalFinances/>,
<https://github.com/R4GoodAcademy/R4GoodPersonalFinances>

BugReports <https://github.com/R4GoodAcademy/R4GoodPersonalFinances/issues>

Depends R (>= 4.1.0)

Imports dplyr, ggplot2, ggrepel, ggtext, glue, withr, PrettyCols, scales, tidyr, shiny, bslib, bsicons

Suggests testthat (>= 3.0.0), vdiffr

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository <https://r4goodacademy.r-universe.dev>

RemoteUrl <https://github.com/R4GoodAcademy/R4GoodPersonalFinances>

RemoteRef HEAD

RemoteSha e4d79c472e68066de48515551ac1ff707176a762

Contents

calc_optimal_risky_asset_allocation	2
calc_purchasing_power	3
calc_risk_adjusted_return	4
plot_purchasing_power	5
plot_risk_adjusted_returns	6
print_currency	7
run_app	8

Index	9
--------------	----------

calc_optimal_risky_asset_allocation
Calculate optimal risky asset allocation

Description

Calculates the optimal allocation to the risky asset using the Merton Share formula.

Usage

```
calc_optimal_risky_asset_allocation(
  risky_asset_return_mean,
  risky_asset_return_sd,
  safe_asset_return,
  risk_aversion
)
```

Arguments

risky_asset_return_mean
 A numeric. The expected (average) yearly return of the risky asset.

risky_asset_return_sd
 A numeric. The standard deviation of the yearly returns of the risky asset.

safe_asset_return
 A numeric. The expected yearly return of the safe asset.

risk_aversion A numeric. The risk aversion coefficient.

Details

Can be used to calculate the optimal allocation to the risky asset for vectors of inputs.

Value

A numeric. The optimal allocation to the risky asset. In case of `NaN()` (because of division by zero) the optimal allocation to the risky asset is set to 0.

See Also

- [How to Determine Our Optimal Asset Allocation?](#)
- Haghani V., White J. (2023) "The Missing Billionaires: A Guide to Better Financial Decisions." ISBN:978-1-119-74791-8.

Examples

```
calc_optimal_risky_asset_allocation(  
  risky_asset_return_mean = 0.05,  
  risky_asset_return_sd   = 0.15,  
  safe_asset_return       = 0.02,  
  risk_aversion           = 2  
)  
  
calc_optimal_risky_asset_allocation(  
  risky_asset_return_mean = c(0.05, 0.06),  
  risky_asset_return_sd   = c(0.15, 0.16),  
  safe_asset_return       = 0.02,  
  risk_aversion           = 2  
)
```

calc_purchasing_power *Calculate purchasing power*

Description

Calculates changes in purchasing power over time, taking into account the real interest rate.

Usage

```
calc_purchasing_power(x, years, real_interest_rate)
```

Arguments

x	A numeric. The initial amount of money.
years	A numeric. The number of years.
real_interest_rate	A numeric. The yearly real interest rate.

Details

The real interest rate is the interest rate after inflation. If negative (e.g. equal to the average yearly inflation rate) it can show diminishing purchasing power over time. If positive, it can show increasing purchasing power over time, and effect of compounding interest on the purchasing power.

Value

A numeric. The purchasing power.

See Also

- [How to Determine Our Optimal Asset Allocation?](#)

Examples

```
calc_purchasing_power(x = 10, years = 30, real_interest_rate = -0.02)
calc_purchasing_power(x = 10, years = 30, real_interest_rate = 0.02)
```

```
calc_risk_adjusted_return
    Calculate risk adjusted return
```

Description

Calculates the risk adjusted return for portfolio of given allocation to the risky asset.

Usage

```
calc_risk_adjusted_return(  
  safe_asset_return,  
  risky_asset_return_mean,  
  risky_asset_allocation,  
  risky_asset_return_sd = NULL,  
  risk_aversion = NULL  
)
```

Arguments

```
safe_asset_return  
    A numeric. The expected yearly return of the safe asset.  
risky_asset_return_mean  
    A numeric. The expected (average) yearly return of the risky asset.  
risky_asset_allocation  
    A numeric. The allocation to the risky asset. Could be a vector. If it is the optimal allocation then parameters risky_asset_return_sd and risk_aversion can be omitted.  
risky_asset_return_sd  
    A numeric. The standard deviation of the yearly returns of the risky asset.  
risk_aversion  
    A numeric. The risk aversion coefficient.
```

Value

A numeric. The risk adjusted return.

See Also

- [How to Determine Our Optimal Asset Allocation?](#)
- Haghani V., White J. (2023) "The Missing Billionaires: A Guide to Better Financial Decisions." ISBN:978-1-119-74791-8.

Examples

```
calc_risk_adjusted_return(
  safe_asset_return = 0.02,
  risky_asset_return_mean = 0.04,
  risky_asset_return_sd = 0.15,
  risky_asset_allocation = 0.5,
  risk_aversion = 2
)

calc_risk_adjusted_return(
  safe_asset_return = 0.02,
  risky_asset_return_mean = 0.04,
  risky_asset_allocation = c(0.25, 0.5, 0.75),
  risky_asset_return_sd = 0.15,
  risk_aversion = 2
)
```

plot_purchasing_power *Plotting changes to the purchasing power over time*

Description

Plots the effect of real interest rates (positive or negative) on the purchasing power of savings over the span of 50 years (default).

Usage

```
plot_purchasing_power(
  x,
  real_interest_rate,
  years = 50,
  legend_title = "Real interest rate",
  seed = NA
)
```

Arguments

x	A numeric. The initial amount of money.
real_interest_rate	A numeric. The yearly real interest rate.
years	A numeric. The number of years.
legend_title	A character.
seed	A numeric. Seed passed to geom_label_repel().

Value

A `ggplot2::ggplot()` object.

See Also

- [How to Determine Our Optimal Asset Allocation?](#)

Examples

```
plot_purchasing_power(  
  x = 10,  
  real_interest_rate = seq(-0.02, 0.04, by = 0.02)  
)
```

plot_risk_adjusted_returns

Plotting risk adjusted returns

Description

Plots the risk adjusted returns for portfolios of various allocations to the risky asset.

Usage

```
plot_risk_adjusted_returns(  
  safe_asset_return,  
  risky_asset_return_mean,  
  risky_asset_return_sd,  
  risk_aversion = 2,  
  current_risky_asset_allocation = NULL  
)
```

Arguments

safe_asset_return

A numeric. The expected yearly return of the safe asset.

risky_asset_return_mean

A numeric. The expected (average) yearly return of the risky asset.

risky_asset_return_sd

A numeric. The standard deviation of the yearly returns of the risky asset.

risk_aversion A numeric. The risk aversion coefficient.

current_risky_asset_allocation

A numeric. The current allocation to the risky asset. For comparison with the optimal allocation.

Value

A `ggplot2::ggplot()` object.

See Also

- [How to Determine Our Optimal Asset Allocation?](#)
- Haghani V., White J. (2023) "The Missing Billionaires: A Guide to Better Financial Decisions." ISBN:978-1-119-74791-8.

Examples

```
plot_risk_adjusted_returns(  
  safe_asset_return      = 0.02,  
  risky_asset_return_mean = 0.04,  
  risky_asset_return_sd   = 0.15,  
  risk_aversion          = 2,  
  current_risky_asset_allocation = 0.8  
)
```

print_currency	<i>Printing currency values or percentages</i>
----------------	--

Description

Wrapper functions for printing nicely formatted values.

Usage

```
print_currency(  
  x,  
  suffix = "",  
  big.mark = ",",  
  accuracy = NULL,  
  prefix = NULL,  
  ...  
)  
  
print_percent(x, accuracy = 0.1, ...)
```

Arguments

x	A numeric vector
big.mark	Character used between every 3 digits to separate thousands.
accuracy	A number to round to. Use (e.g.) 0.01 to show 2 decimal places of precision. If NULL, the default, uses a heuristic that should ensure breaks have the minimum number of digits needed to show the difference between adjacent values. Applied to rescaled data.

prefix, suffix Symbols to display before and after value.
 ... Other arguments passed on to `base::format()`.

See Also

`scales::dollar()`
`scales::percent()`

Examples

```
print_currency(2345678, suffix = " PLN")
print_percent(0.52366)
```

run_app	<i>Run a package app</i>
---------	--------------------------

Description

Run a package app

Usage

```
run_app(
  which = c("risk-adjusted-returns", "purchasing-power"),
  res = 120,
  shinylive = FALSE
)
```

Arguments

which	A character. The name of the app to run. Currently available: <ul style="list-style-type: none"> • risk-adjusted-returns - Plotting risk-adjusted returns for various allocations to the risky asset allows you to find the optimal allocation. • purchasing-power - Plotting the effect of real interest rates (positive or negative) on the purchasing power of savings over time.
res	A numeric. The initial resolution of the plots.
shinylive	A logical. Whether to use shinylive for the app.

Index

`base::format()`, 8

`calc_optimal_risky_asset_allocation`, 2

`calc_purchasing_power`, 3

`calc_risk_adjusted_return`, 4

`ggplot2::ggplot()`, 6, 7

`NaN()`, 2

`plot_purchasing_power`, 5

`plot_risk_adjusted_returns`, 6

`print_currency`, 7

`print_percent(print_currency)`, 7

`run_app`, 8

`scales::dollar()`, 8

`scales::percent()`, 8