# Package 'riskParityPortfolio'

June 1, 2021

**Title** Design of Risk Parity Portfolios

**Version** 0.2.2

**Date** 2021-05-31

**Description** Fast design of risk parity portfolios for financial investment.
The goal of the risk parity portfolio formulation is to equalize or distribute
the risk contributions of the different assets, which is missing if we simply
consider the overall volatility of the portfolio as in the mean-variance
Markowitz portfolio. In addition to the vanilla formulation, where the risk
contributions are perfectly equalized subject to no shortselling and budget
constraints, many other formulations are considered that allow for box
constraints and shortselling, as well as the inclusion of additional
objectives like the expected return and overall variance. See vignette for
a detailed documentation and comparison, with several illustrative examples.
The package is based on the papers:
Y. Feng, and D. P. Palomar (2015). SCRIP: Successive Convex Optimization Methods
for Risk Parity Portfolio Design. IEEE Trans. on Signal Processing, vol. 63,
no. 19, pp. 5285-5300. <doi:10.1109/TSP.2015.2452219>.
F. Spinu (2013), An Algorithm for Computing Risk Parity Weights.
<doi:10.2139/ssrn.2297383>.
T. Griveau-Billion, J. Richard, and T. Roncalli (2013). A fast algorithm for computing
High-dimensional risk parity portfolios. <arXiv:1311.4057>.

**Maintainer** Daniel P. Palomar <daniel.p.palomar@gmail.com>

**URL** https://CRAN.R-project.org/package=riskParityPortfolio,
https://github.com/dppalomar/riskParityPortfolio,
https://www.danielppalomar.com,
https://doi.org/10.1109/TSP.2015.2452219

**BugReports** https://github.com/dppalomar/riskParityPortfolio/issues

**License** GPL-3

**Encoding** UTF-8

**LinkingTo** Rcpp, RcppEigen

**RoxygenNote** 7.1.1

**Imports** alabama, Matrix, nloptr, quadprog, Rcpp

**Suggests** knitr, ggplot2, numDeriv, portfolioBacktest, prettydoc,
rmarkdown, R.rsp, testthat, viridisLite

**VignetteBuilder** knitr, rmarkdown, R.rsp

**NeedsCompilation** yes

**Author** Ze Vinicius [aut],
Daniel P. Palomar [cre, aut]

**Repository** CRAN

**Date/Publication** 2021-06-01 05:50:02 UTC

# R topics documented:

---

riskParityPortfolio-package

*riskParityPortfolio: Design of Risk Parity Portfolios*

---

#### Description

Fast design of risk parity portfolios for financial investment. The goal of the risk parity portfolio formulation is to equalize or distribute the risk contributions of the different assets, which is missing if we simply consider the overall volatility of the portfolio as in the mean-variance Markowitz portfolio. In addition to the vanilla formulation, where the risk contributions are perfectly equalized subject to no shortselling and budget constraints, many other formulations are considered that allow for box constraints and shortselling, as well as the inclusion of additional objectives like the expected return and overall variance. See vignette for a detailed documentation and comparison, with several illustrative examples.

#### Functions

riskParityPortfolio, barplotPortfolioRisk

#### Help

For a quick help see the README file: GitHub-README.

For more details see the vignette: CRAN-vignette.

#### Author(s)

Ze Vinicius and Daniel P. Palomar

## References

Y. Feng, and D. P. Palomar (2015). SCRIP: Successive Convex Optimization Methods for Risk Parity Portfolio Design. *IEEE Trans. on Signal Processing*, vol. 63, no. 19, pp. 5285-5300. <https://doi.org/10.1109/TSP.2015.2452219>

F. Spinu (2013). An Algorithm for Computing Risk Parity Weights. <https://dx.doi.org/10.2139/ssrn.2297383>

T. Griveau-Billion, J. Richard, and T. Roncalli (2013). A fast algorithm for computing High-dimensional risk parity portfolios. <https://arxiv.org/pdf/1311.4057.pdf>

---

| barplotPortfolioRisk | *Create portfolio barplots with the capital allocation and the risk allocation* |
|---|---|

---

## Description

Creates a barplot on top with the portfolio capital allocation and another at the bottom with the risk contribution allocation whose profile is the target of the risk parity portfolio design with riskParityPortfolio. By default the plot is based on the package ggplot2, but the user can also specify a simple base plot.

## Usage

```
barplotPortfolioRisk(w, Sigma, type = c("ggplot2", "simple"), colors = NULL)
```

## Arguments

| | |
|---|---|
| w | Vector or matrix containing the portfolio(s) weights. For multiple portfolios, they should be columnwise and named for the legend. |
| Sigma | Covariance matrix of the assets. |
| type | Type of plot. Valid options: "ggplot2","simple". Default is "ggplot2" (the packages ggplot2 and gridExtra must be installed). |
| colors | Vector of colors for the portfolios (optional). |

## Author(s)

Daniel P. Palomar and Ze Vinicius

## Examples

```
library(riskParityPortfolio)

# generate random covariance matrix
set.seed(42)
N <- 10
V <- matrix(rnorm(N^2), nrow = N)
Sigma <- cov(V)
```

```
# generate random portfolio vectors
w_single <- runif(N)
w_single <- w_single/sum(w_single)  # normalize
names(w_single) <- LETTERS[1:N]

w_multiple <- matrix(runif(4*N), ncol = 4)
w_multiple <- sweep(w_multiple,  # normalize each column
                    MARGIN = 2,
                    STATS = colSums(w_multiple), FUN = "/")
rownames(w_multiple) <- LETTERS[1:N]

# plot
barplotPortfolioRisk(w_single, Sigma)
barplotPortfolioRisk(w_multiple, Sigma)
barplotPortfolioRisk(w_multiple, Sigma, colors = viridisLite::viridis(4))
barplotPortfolioRisk(w_multiple, Sigma) + ggplot2::scale_fill_viridis_d()
```

---

riskParityPortfolio          *Design of risk parity portfolios*

---

### Description

This function designs risk parity portfolios to equalize/distribute the risk contributions of the different assets, which is missing if we simply consider the overall volatility of the portfolio as in the mean-variance Markowitz portfolio. In addition to the vanilla formulation, where the risk contributions are perfectly equalized subject to no shortselling and budget constraints, many other formulations are considered that allow for box constraints, as well as the inclusion of additional objectives like the expected return and overall variance. In short, this function solves the following problem:

minimize R(w) -lmd_mu * t(w) %*% mu + lmd_var * t(w) %*% Sigma %*% w

subject to sum(w) = 1,w_lb <= w <= w_ub, Cmat %*% w = cvec,Dmat %*% w <= dvec,

where R(w) denotes the risk concentration, t(w) %*% mu is the expected return, t(w) %*% Sigma %*% w is the overall variance, lmd_mu and lmd_var are the trade-off weights for the expected return and the variance terms, respectively, w_lb and w_ub are the lower and upper bound vector values for the portfolio vector w, Cmat %*% w = cvec denotes arbitrary linear equality constrains, and Dmat %*% w = dvec denotes arbitrary linear inequality constrains.

### Usage

```
riskParityPortfolio(
  Sigma,
  b = NULL,
  mu = NULL,
  lmd_mu = 0,
  lmd_var = 0,
  w_lb = 0,
  w_ub = 1,
```

```
    Cmat = NULL,
    cvec = NULL,
    Dmat = NULL,
    dvec = NULL,
    method_init = c("cyclical-spinu", "cyclical-roncalli", "newton"),
    method = c("sca", "alabama", "slsqp"),
    formulation = NULL,
    w0 = NULL,
    theta0 = NULL,
    gamma = 0.9,
    zeta = 1e-07,
    tau = NULL,
    maxiter = 1000,
    ftol = 1e-08,
    wtol = 5e-07,
    use_gradient = TRUE,
    use_qp_solver = TRUE
)
```

## Arguments

| | |
|---|---|
| Sigma | Covariance or correlation matrix (this is the only mandatory argument). |
| b | Budget vector, i.e., the risk budgeting targets. The default is the uniform 1/N vector. |
| mu | Vector of expected returns (only needed if the expected return term is desired in the objective). |
| lmd_mu | Scalar weight to control the importance of the expected return term. |
| lmd_var | Scalar weight to control the importance of the variance term (only currently available for the SCA method). |
| w_lb | Lower bound (either a vector or a scalar) on the value of each portfolio weight. |
| w_ub | Upper bound (either a vector or a scalar) on the value of each portfolio weight. |
| Cmat | Equality constraints matrix. |
| cvec | Equality constraints vector. |
| Dmat | Inequality constraints matrix. |
| dvec | Inequality constraints vector. |
| method_init | Method to compute the vanilla solution. In case of additional constraints or objective terms, this solution is used as the initial point for the subsequent method. The default is "cyclical-spinu". See details below. |
| method | Method to solve the non-vanilla formulation. The default is "sca". See details below. (DEPRECATED) |
| formulation | String indicating the risk concentration formulation to be used. It must be one of: "diag", "rc-double-index", "rc-over-b-double-index", "rc-over-var vs b", "rc-over-var", "rc-over-sd vs b-times-sd", "rc vs b-times-var", "rc vs theta", or "rc-over-b vs theta". The default is "rc-over-b-double-index". If formulation is not provided and no additional terms or constraints are set, such as expected |

|          | return or shortselling, then the vanilla risk parity portfolio will be returned. If formulation is "diag" then the analytical solution of the risk parity optimization for for a diagonal covariance matrix will be returned. See details below. |
|----------|------|
| w0       | Initial value for the portfolio weights. Default is a convex combination of the risk parity portfolio, the (uncorrelated) minimum variance portfolio, and the maximum return portfolio. |
| theta0   | Initial value for theta (in case formulation uses theta). If not provided, the optimum solution for a fixed vector of portfolio weights will be used. |
| gamma    | Learning rate for the SCA method. |
| zeta     | Factor used to decrease the learning rate at each iteration for the SCA method. |
| tau      | Regularization factor. |
| maxiter  | Maximum number of iterations for the SCA loop. |
| ftol     | Convergence tolerance on the objective function. |
| wtol     | Convergence tolerance on the values of the portfolio weights. |
| use_gradient | This parameter is meaningful only if method is either "alabama" or "slsqp". If TRUE (default value), analytical gradients of the objective function will be used (strongly recommended to achieve faster results). |
| use_qp_solver | Whether or not to use the general QP solver from quadprog to solve each iteration of the SCA algorithm. Default is TRUE. |

### Details

By default, the problem considered is the vanilla risk parity portfolio: `w >= 0, sum(w) = 1`, with no expected return term, and no variance term. In this case, the problem formulation is convex and the optimal solution is guaranteed to be achieved with a perfect risk concentration, i.e., `R(w) = 0`. By default, we use the formulation by Spinu (2013) (`method_init = "cyclical-spinu"`), but the user can also select the formulation by Roncalli et al. (2013) (`method_init = "cyclical-roncalli"`).

In case of additional box constraints, expected return term, or variance term, then the problem is nonconvex and the global optimal solution cannot be guaranteed, just a local optimal. We use the efficient sucessive convex approximation (SCA) method proposed in Feng & Palomar (2015), where the user can choose among many different risk concentration terms (through the argument `formulation`), namely:

- `formulation = "rc-double-index"`: $\sum_{i,j} (r_i - r_j)^2$
- `formulation = "rc-vs-theta"`: $\sum_{i} (r_i - theta)^2$
- `formulation = "rc-over-var-vs-b"`: $\sum_{i} (r_i/r - b_i)^2$
- `formulation = "rc-over-b double-index"`: $\sum_{i,j} (r_i/b_i - r_j/b_j)^2$
- `formulation = "rc-vs-b-times-var"`: $\sum_{i} (r_i - b_i*r)^2$
- `formulation = "rc-over-sd vs b-times-sd"`: $\sum_{i} (r_i/\sqrt{r} - b_i*\sqrt{r})^2$
- `formulation = "rc-over-b vs theta"`: $\sum_{i} (r_i/b_i - theta)^2$
- `formulation = "rc-over-var"`: $\sum_{i} (r_i/r)^2$

where `r_i = w_i*(Sigma%*%w)_i` is the risk contribution and `r = t(w)%*%Sigma%*%w` is the overall risk (i.e., variance).

For more details, please check the vignette.

## Value

A list containing possibly the following elements:

w                  Optimal portfolio vector.

relative_risk_contribution
                   The relative risk contribution of every asset.

theta              Optimal value for theta (in case that it is part of the chosen formulation.)

obj_fun            Sequence of values of the objective function at each iteration.

risk_concentration
                   Risk concentration term of the portfolio R(w).

mean_return        Expected return term of the portoflio t(w)%*%mu, if the term is included in the optimization.

variance           Variance term of the portfolio t(w)%*%Sigma%*%w, if the term is included in the optimization.

elapsed_time       Elapsed time recorded at every iteration.

convergence        Boolean flag to indicate whether or not the optimization converged.

is_feasible        Boolean flag to indicate whether or not the computed portfolio respects the linear constraints.

## Author(s)

Ze Vinicius and Daniel P. Palomar

## References

Y. Feng, and D. P. Palomar (2015). SCRIP: Successive Convex Optimization Methods for Risk Parity Portfolio Design. *IEEE Trans. on Signal Processing*, vol. 63, no. 19, pp. 5285-5300. <https://doi.org/10.1109/TSP.2015.2452219>

F. Spinu (2013). An Algorithm for Computing Risk Parity Weights. <https://dx.doi.org/10.2139/ssrn.2297383>

T. Griveau-Billion, J. Richard, and T. Roncalli (2013). A fast algorithm for computing High-dimensional risk parity portfolios. <https://arxiv.org/pdf/1311.4057.pdf>

## Examples

```
library(riskParityPortfolio)

# create covariance matrix
N <- 5
V <- matrix(rnorm(N^2), ncol = N)
Sigma <- cov(V)

# risk parity portfolio
res <- riskParityPortfolio(Sigma)
names(res)
#> [1] "w"                 "risk_contribution"

res$w
```

```
#> [1] 0.04142886 0.38873465 0.34916787 0.09124019 0.12942842

res$relative_risk_contribution
#> [1] 0.2 0.2 0.2 0.2 0.2

# risk budggeting portfolio
res <- riskParityPortfolio(Sigma, b = c(0.4, 0.4, 0.1, 0.05, 0.05))
res$relative_risk_contribution
#> [1] 0.40 0.40 0.10 0.05 0.05
```

# Index