Complex Portfolio Optimization with PortfolioAnalytics

R/Finance 2014

Ross Bennett

Overview

- Discuss Portfolio Optimization
- Introduce PortfolioAnalytics
- Demonstrate PortfolioAnalytics with Examples

Modern Portfolio Theory

"Modern" Portfolio Theory (MPT) was introduced by Harry Markowitz in 1952.

In general, MPT states that an investor's objective is to maximize portfolio expected return for a given amount of risk.

General Objectives

- Maximize a measure of gain per unit measure of risk
- Minimize a measure of risk

How do we define risk? What about more complex objectives and constraints?

Portfolio Optimization Objectives

- Minimize Risk
  - Volatility
  - Tail Loss (VaR, ES)
  - Other Downside Risk Measure
- Maximize Risk Adjusted Return
  - Sharpe Ratio, Modified Sharpe Ratio
  - Several Others
- Risk Budgets
  - Equal Component Contribution to Risk (i.e. Risk Parity)
  - Limits on Component Contribution
- Maximize a Utility Function
  - Quadratic, CRRA, etc.

PortfolioAnalytics Overview

PortfolioAnalytics is an R package designed to provide numerical solutions and visualizations for portfolio optimization problems with complex constraints and objectives.

- Support for multiple constraint and objective types
- An objective function can be any valid R function
- Modular constraints and objectives
- Support for user defined moment functions
- Visualizations
- Solver agnostic
- Support for parallel computing

Support Multiple Solvers

Linear and Quadratic Programming Solvers

- R Optimization Infrastructure (ROI)
    - GLPK (Rglpk)
    - Symphony (Rsymphony)
    - Quadprog (quadprog)

Global (stochastic or continuous solvers)

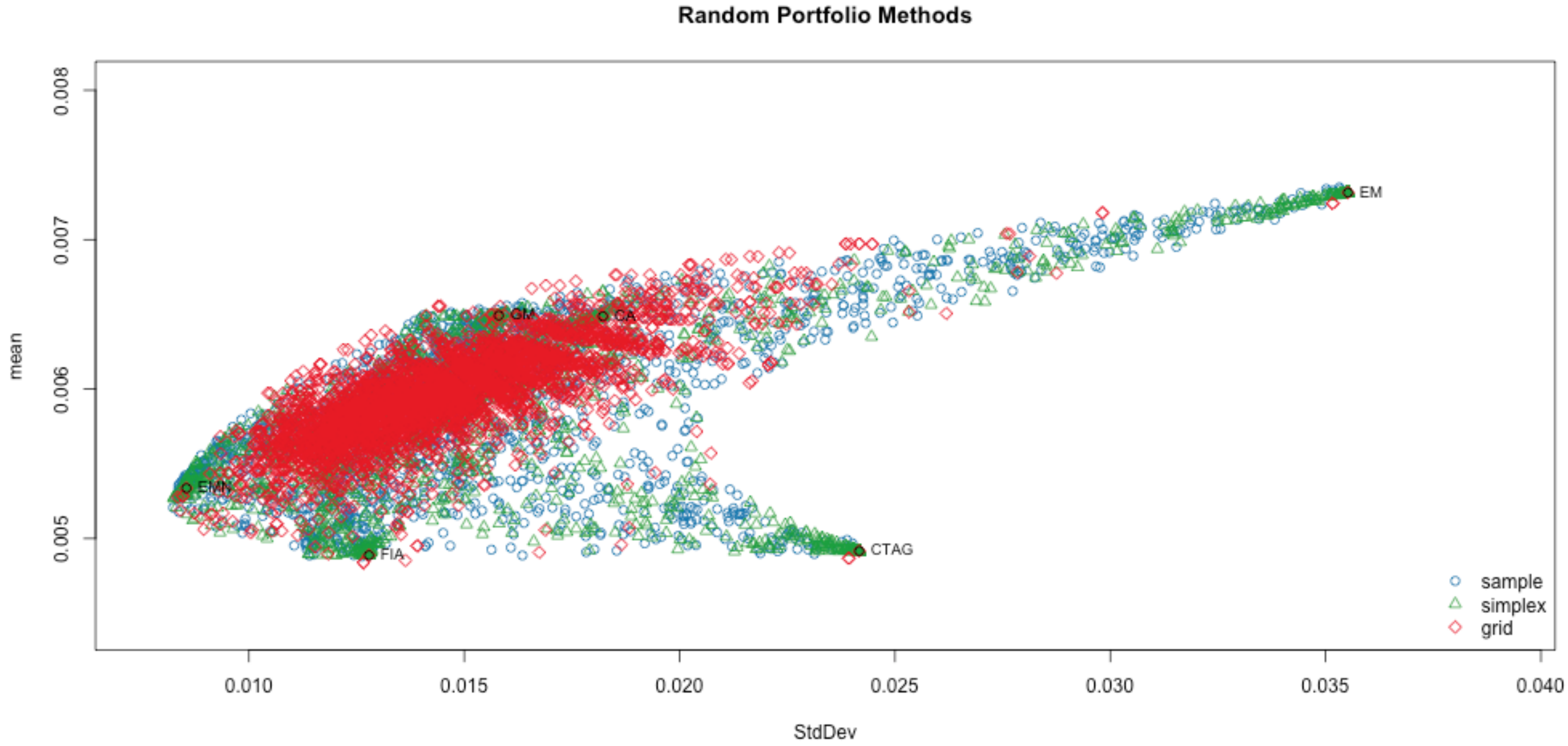
- Random Portfolios
- Differential Evolution (DEoptim)
- Particle Swarm Optimization (pso)
- Generalized Simulated Annealing (GenSA)

Random Portfolios

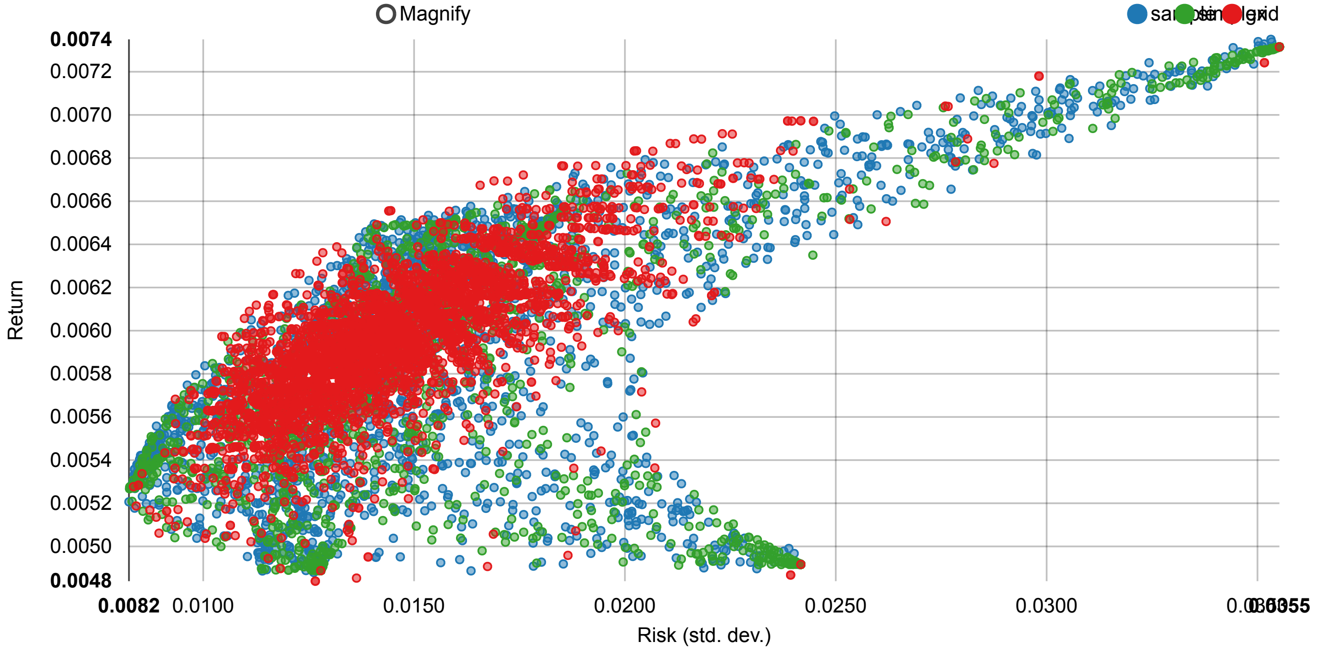
PortfolioAnalytics has three methods to generate random portfolios.

1. The sample method to generate random portfolios is based on an idea by Pat Burns.
2. The simplex method to generate random portfolios is based on a paper by W. T. Shaw.
3. The grid method to generate random portfolios is based on the `gridSearch` function in the NMOF package.

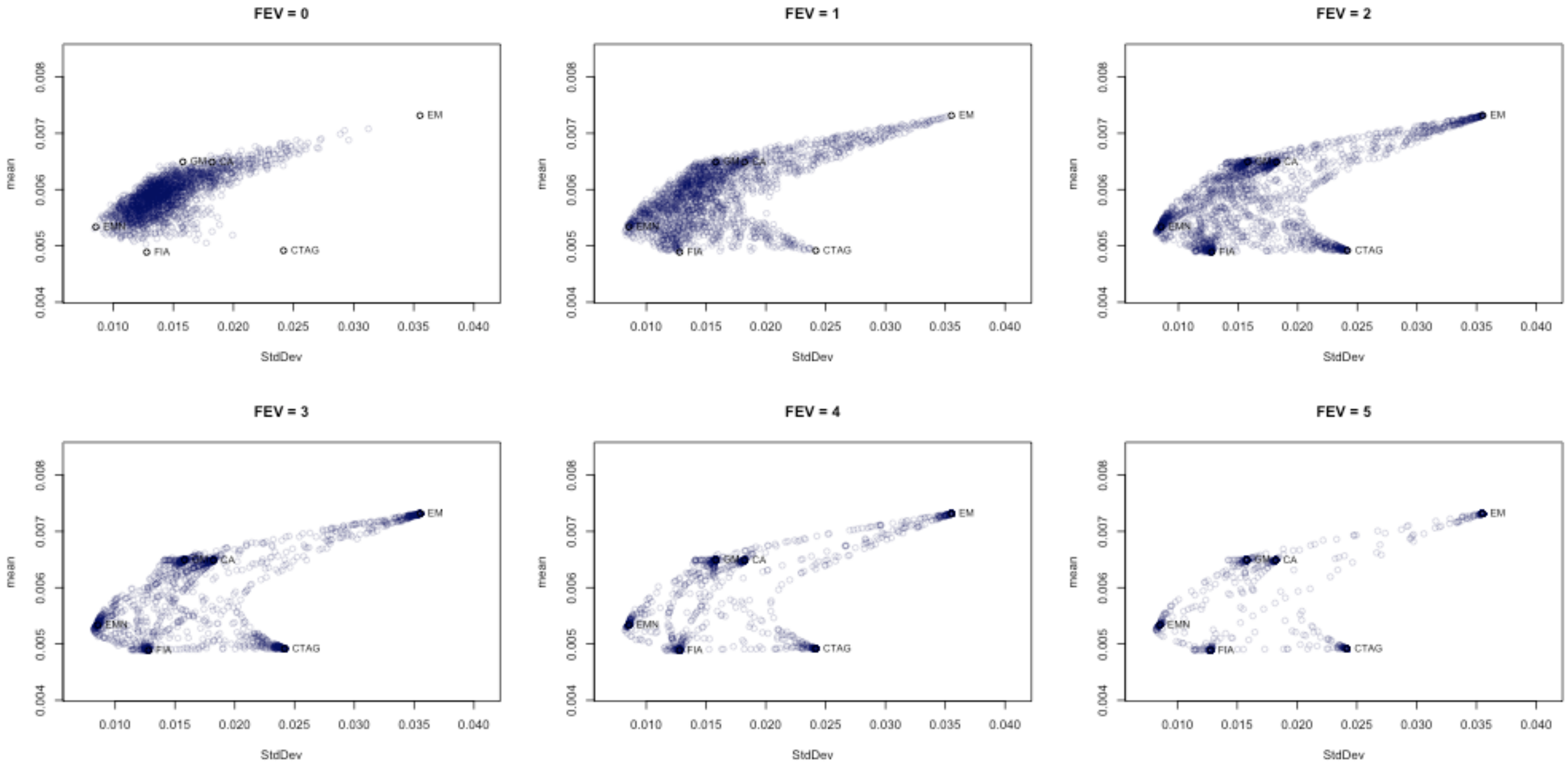
Comparison of Random Portfolio Methods



Comparison of Random Portfolio Methods (Interactive!)



Random Portfolios: Simplex Method



Workflow: Specify Portfolio

```
args(portfolio.spec)
```

```
## function (assets = NULL, category_labels = NULL, weight_seq = NULL,
##     message = FALSE)
## NULL
```

Initializes the portfolio object that holds portfolio level data, constraints, and objectives

Workflow: Add Constraints

```
args(add.constraint)
```

```
## function (portfolio, type, enabled = TRUE, message = FALSE, ...,
##     indexnum = NULL)
## NULL
```

Supported Constraint Types

- Sum of Weights
- Box
- Group
- Factor Exposure
- Position Limit
- and many more

Workflow: Add Objectives

```
args(add.objective)
```

```
## function (portfolio, constraints = NULL, type, name, arguments = NULL,
##     enabled = TRUE, ..., indexnum = NULL)
## NULL
```

Supported Objective types

- Return
- Risk
- Risk Budget
- Weight Concentration

Workflow: Run Optimization

```
args(optimize.portfolio)
```

```
## function (R, portfolio = NULL, constraints = NULL, objectives = NULL,
##     optimize_method = c("DEoptim", "random", "ROI", "pso", "GenSA"),
##     search_size = 20000, trace = FALSE, ..., rp = NULL, momentFUN = "set.portfolio.moments'
##     message = FALSE)
## NULL
```

```
args(optimize.portfolio.rebalancing)
```

```
## function (R, portfolio = NULL, constraints = NULL, objectives = NULL,
##     optimize_method = c("DEoptim", "random", "ROI"), search_size = 20000,
##     trace = FALSE, ..., rp = NULL, rebalance_on = NULL, training_period = NULL,
##     trailing_periods = NULL)
## NULL
```

Workflow: Analyze Results

VISUALIZATION	DATA EXTRACTION
plot	extractObjectiveMeasures
chart.Concentration	extractStats
chart.EfficientFrontier	extractWeights
chart.RiskReward	print
chart.RiskBudget	summary
chart.Weights	

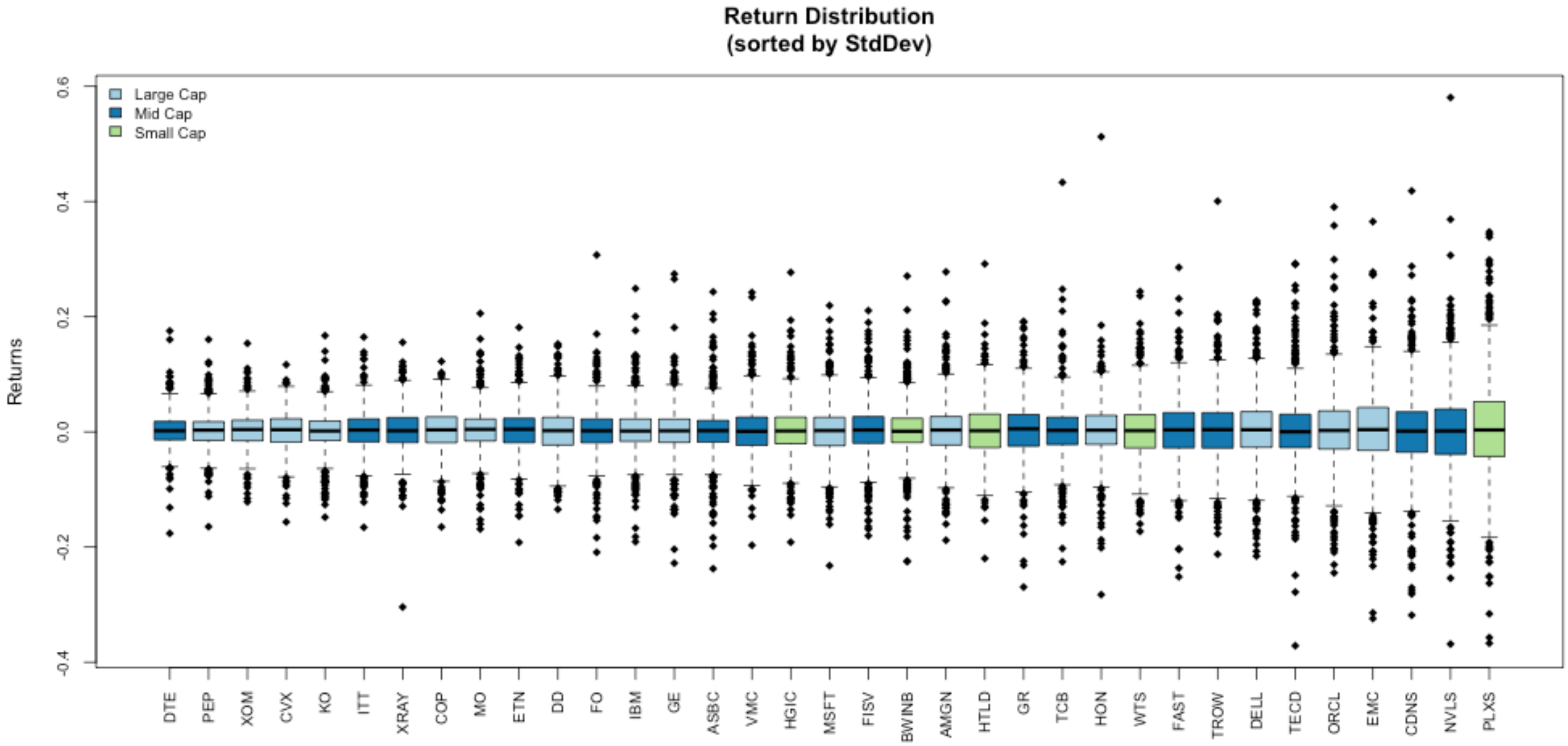
Stock Data Setup

Here we will look at portfolio optimization in the context of stocks.

- Selection of large cap, mid cap, and small cap stocks from CRSP data
- Weekly data from 1/7/1997 to 12/28/2010
- 15 Large Cap
- 15 Mid Cap
- 5 Small Cap

```
equity.data <- cbind(largecap_weekly[,1:15],
                     midcap_weekly[,1:15],
                     smallcap_weekly[,1:5])
```

Distribution of Monthly Returns



Example 1: Market Neutral Portfolio

Here we consider a portfolio of stocks. Our objective is to maximize portfolio return with a target of 0.0015 and minimize portfolio StdDev with a target of 0.02 subject to dollar neutral, beta, box, and position limit constraints.

Specify Portfolio: Constraints

```
portf.dn <- portfolio.spec(stocks)

# Add constraint such that the portfolio weights sum to 0*
portf.dn <- add.constraint(portf.dn, type="weight_sum",
                                  min_sum=-0.01, max_sum=0.01)

# Add box constraint such that no asset can have a weight of greater than
# 20% or less than -20%
portf.dn <- add.constraint(portf.dn, type="box", min=-0.2, max=0.2)

# Add constraint such that we have at most 20 positions
portf.dn <- add.constraint(portf.dn, type="position_limit", max_pos=20)

# Add constraint such that the portfolio beta is between -0.25 and 0.25
betas <- t(CAPM.beta(equity.data, market, Rf))
portf.dn <- add.constraint(portf.dn, type="factor_exposure", B=betas,
                           lower=-0.25, upper=0.25)
```

Specify Portfolio: Objectives

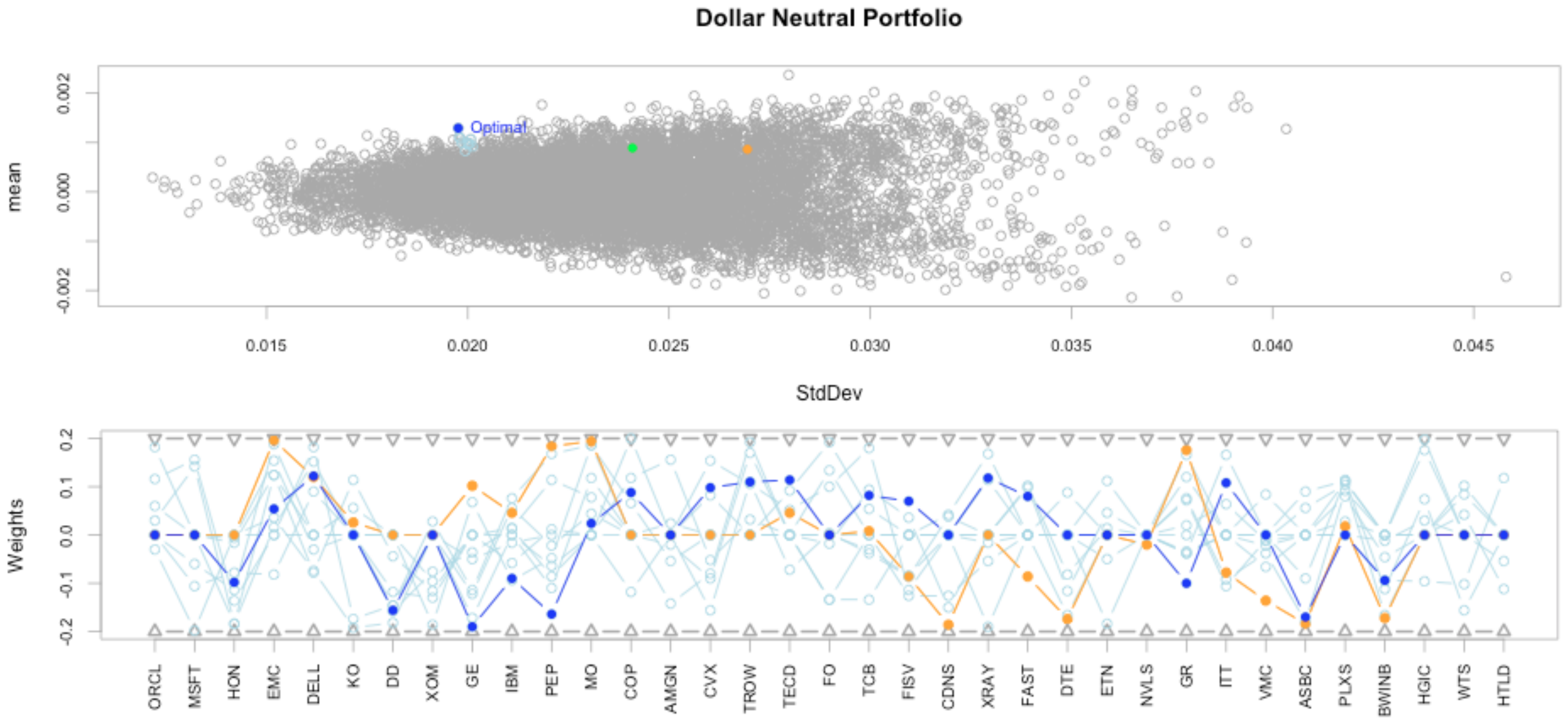
```r
# Add objective to maximize portfolio return with a target of 0.0015
portf.dn.StdDev <- add.objective(portf.dn, type="return", name="mean",
                                 target=0.0015)

# Add objective to minimize portfolio StdDev with a target of 0.02
portf.dn.StdDev <- add.objective(portf.dn.StdDev, type="risk", name="StdDev",
                                 target=0.02)
```

Run Optimization

```
# Generate random portfolios
rp <- random_portfolios(portf.dn, 10000, "sample")

# Run the optimization
opt.dn <- optimize.portfolio(equity.data, portf.dn.StdDev,
                               optimize_method="random", rp=rp,
                               trace=TRUE)
```

Plot Results

```
plot(opt.dn, main="Dollar Neutral Portfolio", risk.col="StdDev", neighbors=10)
```



EDHEC Data Setup

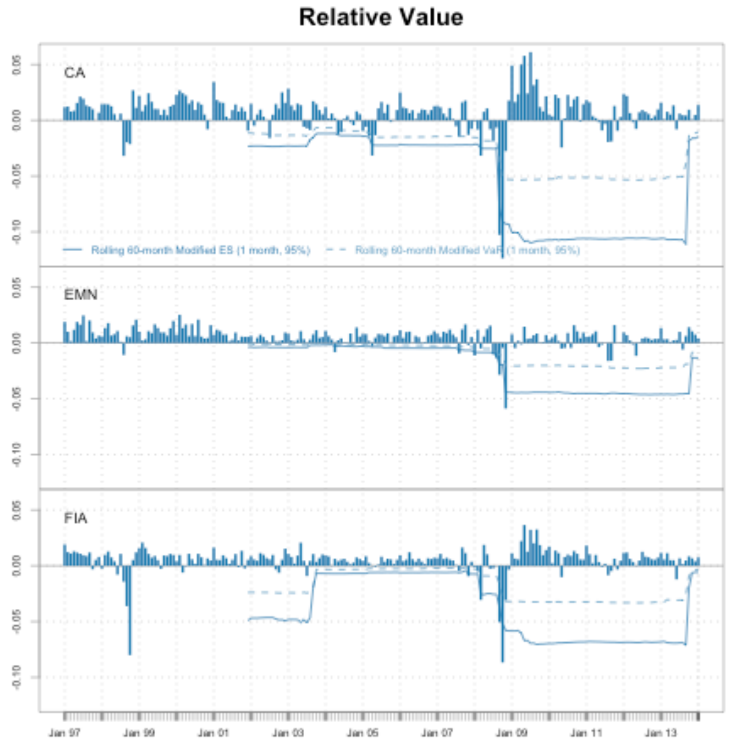
Here we will look at portfolio optimization in the context of portfolio of hedge funds.

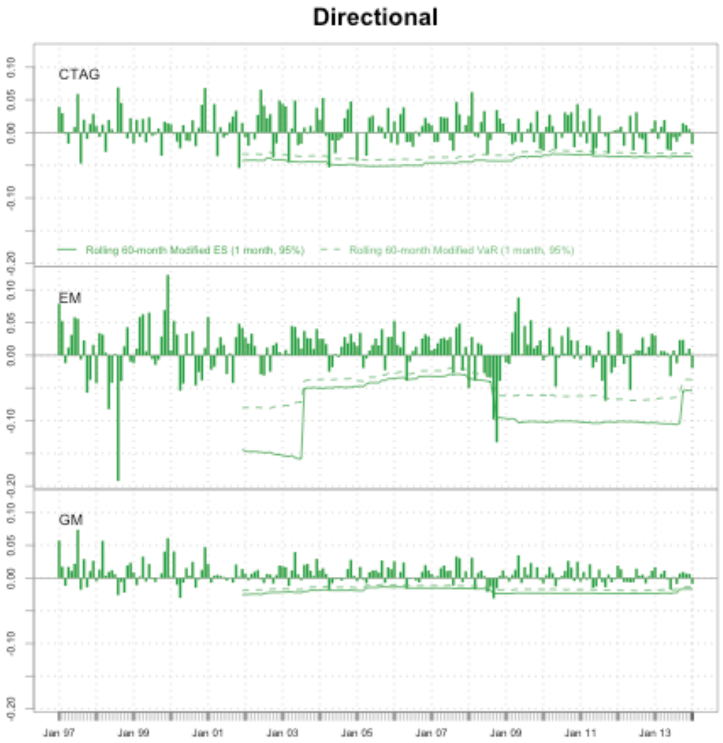
- EDHEC-Risk Alternative Indexes monthly returns from 1/31/1997 to 1/31/2014

RELATIVE VALUE	DIRECTIONAL
Convertible Arbitrage (CA)	CTA Global (CTAG)
Equity Market Neutral (EMN)	Emerging Markets (EM)
Fixed Income Arbitrage (FIA)	Global Macro (GM)

```r
R <- edhec[,c("Convertible.Arbitrage", "Equity.Market.Neutral",
              "Fixed.Income.Arbitrage",
              "CTA.Global", "Emerging.Markets", "Global.Macro")]
# Abreviate column names for convenience and plotting
colnames(R) <- c("CA", "EMN", "FIA", "CTAG", "EM", "GM")
```

Monthly Returns





Example 2: Minimum Expected Shortfall

Consider an allocation to hedge funds using the EDHEC-Risk Alternative Index as a proxy. This will be an extended example starting with an objective to minimize modified expected shortfall, then add risk budget percent contribution limit, and finally add equal risk contribution limit.

- Minimize Modified Expected Shortfall
- Minimize Modified Expected Shortfall with Risk Budget Limit
- Minimize Modified Expected Shortfall with Equal Risk Contribution

Specify Initial Portfolio

```
# Specify an initial portfolio
funds <- colnames(R)
portf.init <- portfolio.spec(funds)

# Add constraint such that the weights sum to 1*
portf.init <- add.constraint(portf.init, type="weight_sum",
                             min_sum=0.99, max_sum=1.01)

# Add box constraint such that no asset can have a weight of greater than
# 40% or less than 5%
portf.init <- add.constraint(portf.init, type="box",
                             min=0.05, max=0.4)

# Add return objective with multiplier=0 such that the portfolio mean
# return is calculated, but does not impact optimization
portf.init <- add.objective(portf.init, type="return",
                            name="mean", multiplier=0)
```

Add Objectives

```r
# Add objective to minimize expected shortfall
portf.minES <- add.objective(portf.init, type="risk", name="ES")

# Add objective to set upper bound on percentage component contribution
portf.minES.RB <- add.objective(portf.minES, type="risk_budget",
                                name="ES", max_prisk=0.3)
# Relax box constraints
portf.minES.RB$constraints[[2]]$max <- rep(1,ncol(R))

# Add objective to minimize concentration of modified ES
# component contribution
portf.minES.EqRB <- add.objective(portf.minES, type="risk_budget",
                                  name="ES", min_concentration=TRUE)
# Relax box constraints
portf.minES.EqRB <- add.constraint(portf.minES.EqRB, type="box",
                                   min=0.05, max=1, indexnum=2)
```

Run Optimization

```r
# Combine the 3 portfolios
portf <- combine.portfolios(list(minES=portf.minES,
                                 minES.RB=portf.minES.RB,
                                 minES.EqRB=portf.minES.EqRB))

# Run the optimization
opt.minES <- optimize.portfolio(R, portf, optimize_method="DEoptim",
                                search_size=5000, trace=TRUE, traceDE=0)
```

Plot in Risk-Return Space

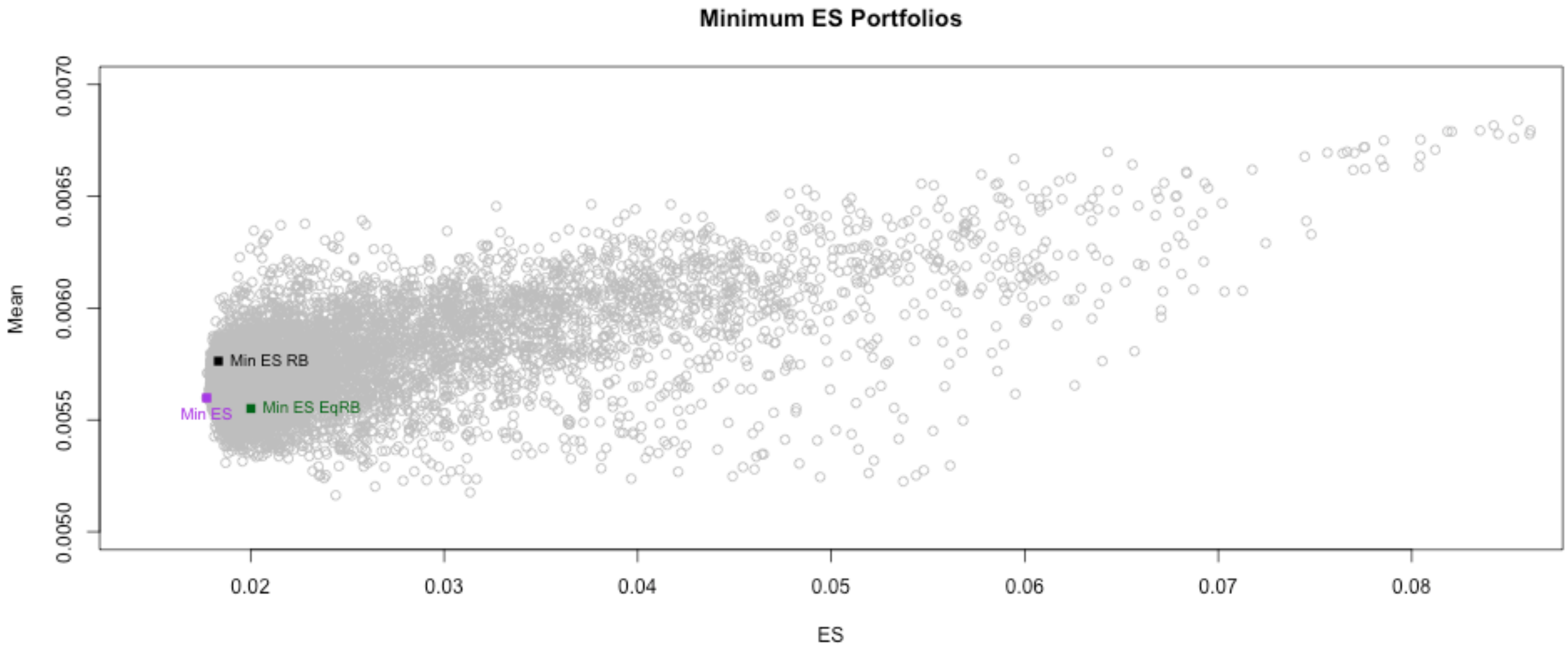
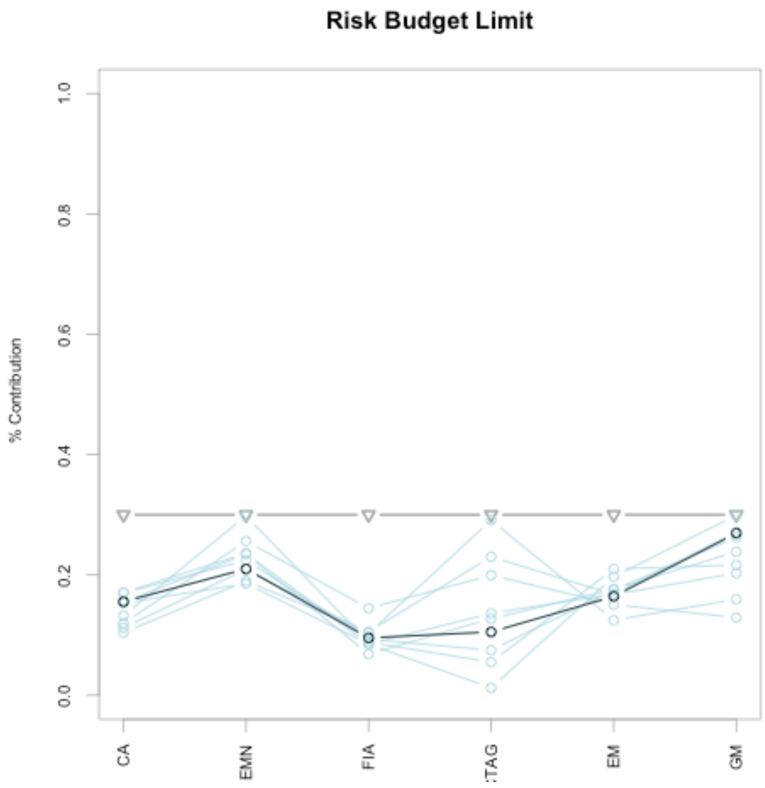
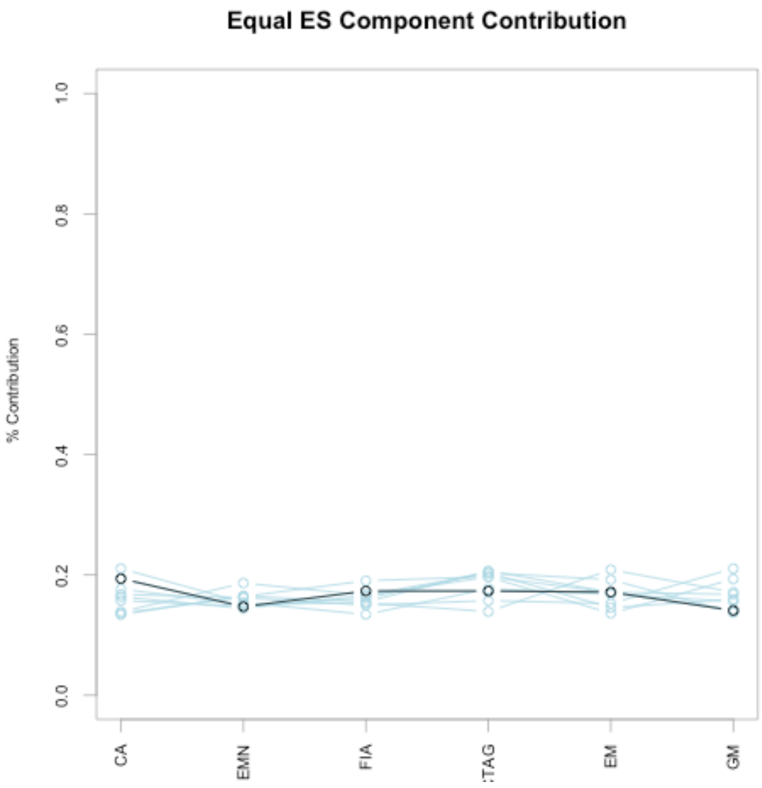


Chart Risk Budgets

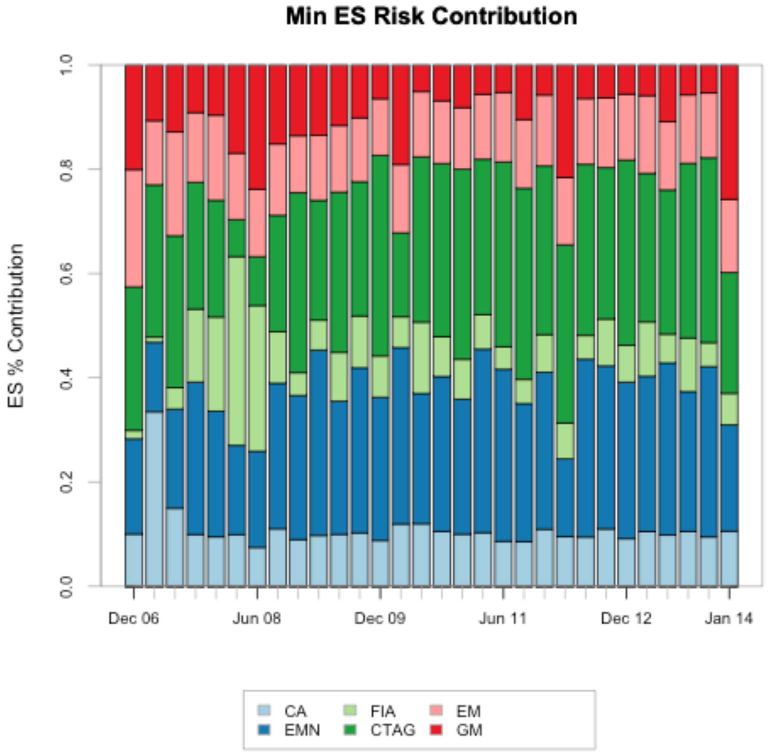

Risk Budget Limit

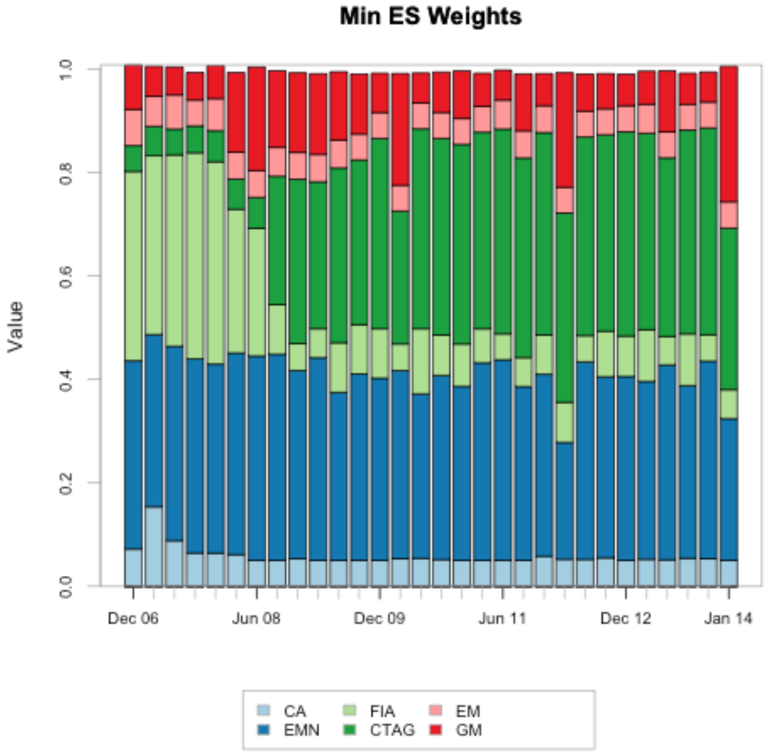

Equal ES Component Contribution

Set Rebalancing Parameters and Run Backtest

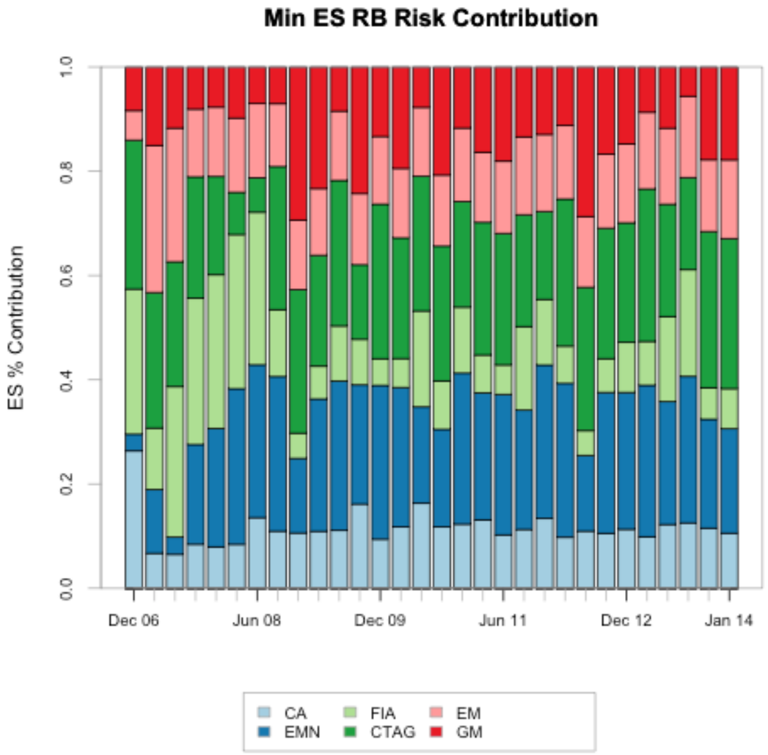
```r
# Set rebalancing frequency
rebal.freq <- "quarters"

# Training Period
training <- 120

# Trailing Period
trailing <- 72

bt.opt.minES <- optimize.portfolio.rebalancing(R, portf,
                                               optimize_method="DEoptim",
                                               rebalance_on=rebal.freq,
                                               training_period=training,
                                               trailing_periods=trailing,
                                               search_size=5000,
                                               traceDE=0)
```

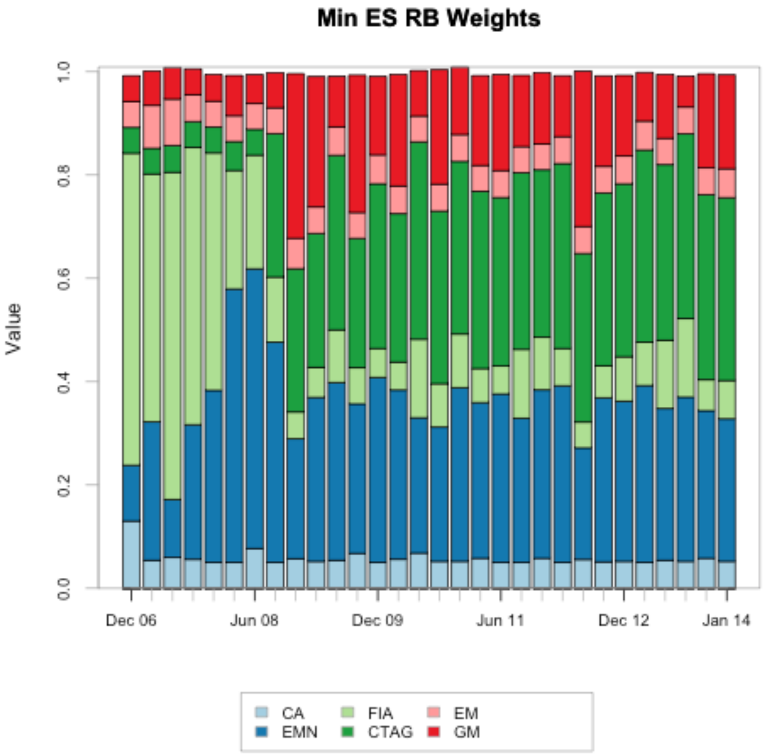
Min ES Risk Contributions and Weights Through Time



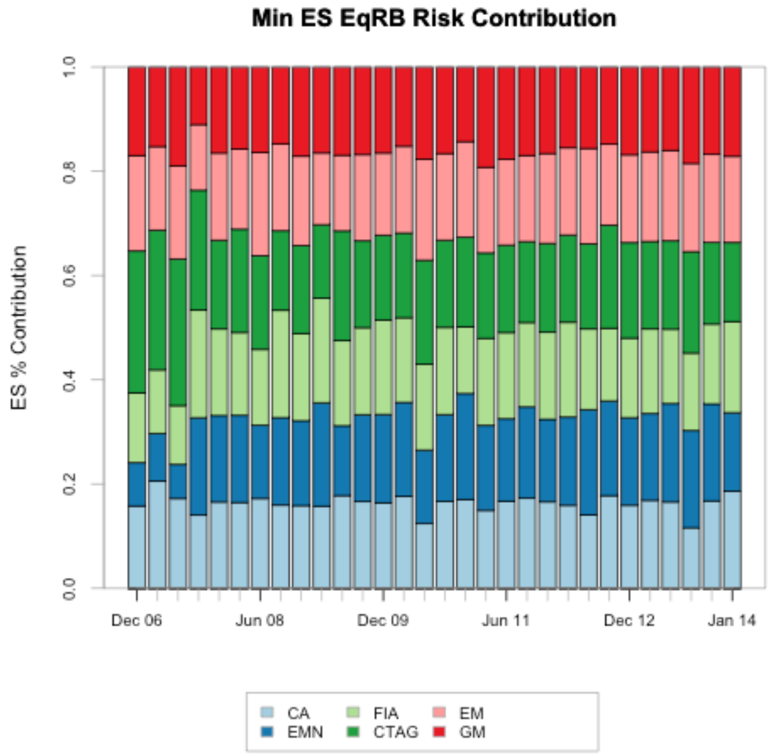


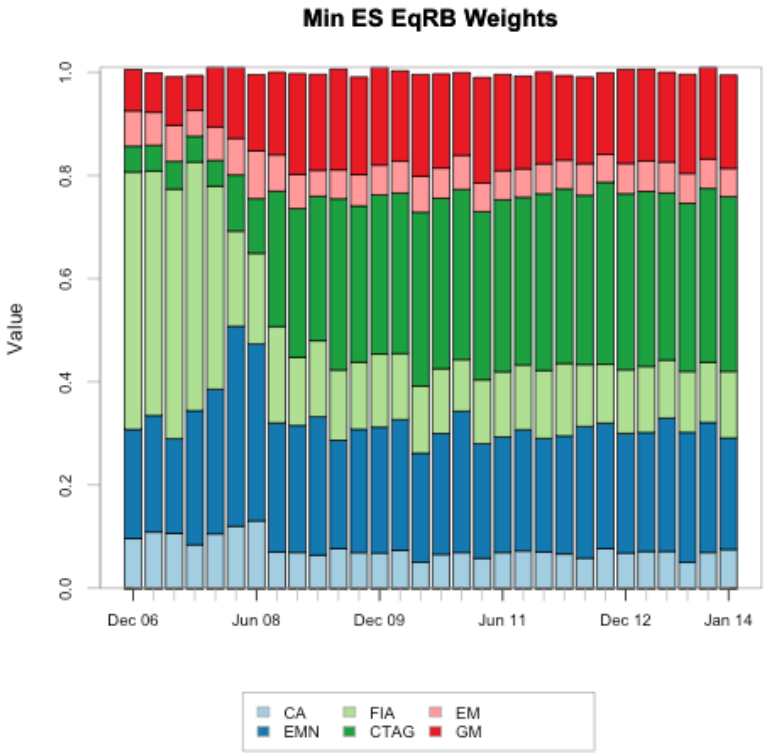
Min ES Risk Budget Limit Risk Contributions and Weights Through Time





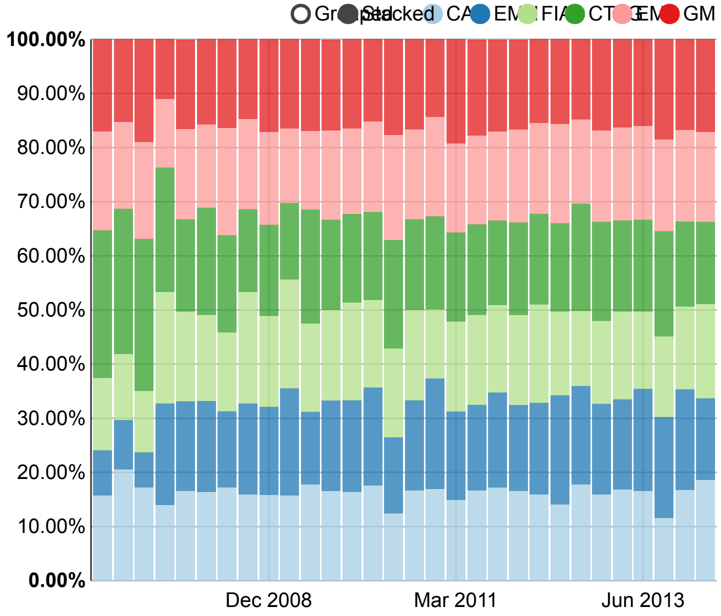
Min ES Equal Component Contribution Risk Contributions and Weights Through Time



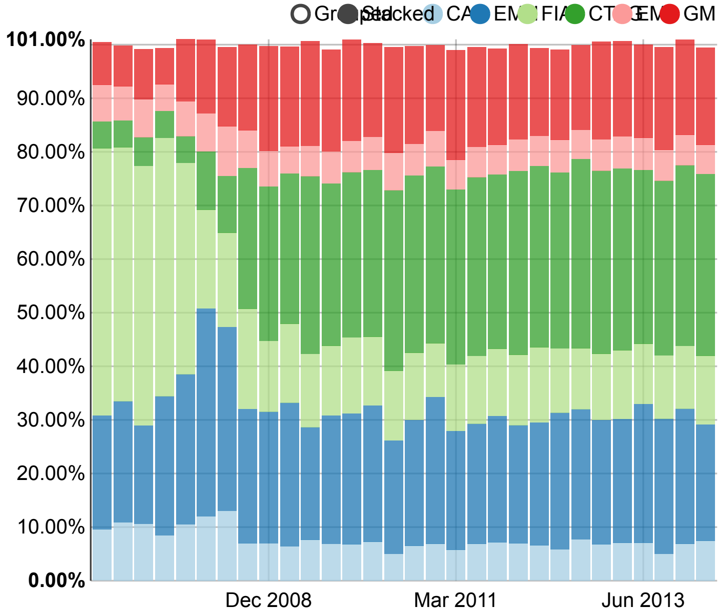


Min ES Equal Component Contribution Risk Contributions and Weights (interactive!)

Percent Contribution to Risk



Weights

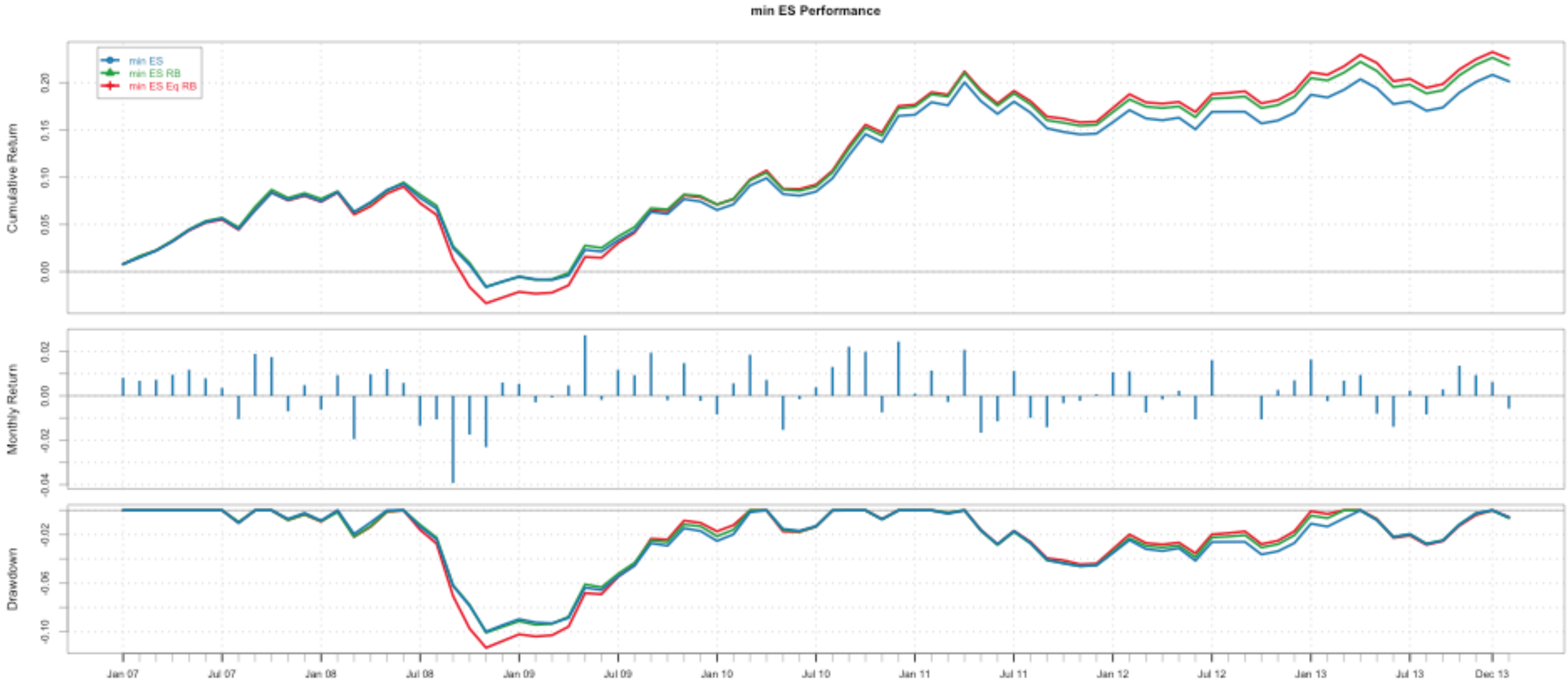


Compute Returns

```r
ret.bt.opt <- do.call(cbind, lapply(bt.opt.minES,
                                    function(x) summary(x)$portfolio_returns))
colnames(ret.bt.opt) <- c("min ES", "min ES RB", "min ES Eq RB")
```

Chart Performance

```
charts.PerformanceSummary(ret.bt.opt)
```



Example 3: Maximize CRRA

Consider an allocation to hedge funds using the EDHEC-Risk Alternative Index as a proxy. Our objective to maximize the fourth order expansion of the Constant Relative Risk Aversion (CRRA) expected utility function as in the Boudt paper and Martellini paper. We use the same data as Example 3.

$$EU_{\lambda}(w) = -\frac{\lambda}{2} m_{(2)}(w) + \frac{\lambda(\lambda+1)}{6} m_{(3)}(w) - \frac{\lambda(\lambda+1)(\lambda+2)}{24} m_{(4)}(w)$$

Define a function to compute CRRA

```r
CRRA <- function(R, weights, lambda, sigma, m3, m4){
  weights <- matrix(weights, ncol=1)
  M2.w <- t(weights) %*% sigma %*% weights
  M3.w <- t(weights) %*% m3 %*% (weights %x% weights)
  M4.w <- t(weights) %*% m4 %*% (weights %x% weights %x% weights)
  term1 <- (1 / 2) * lambda * M2.w
  term2 <- (1 / 6) * lambda * (lambda + 1) * M3.w
  term3 <- (1 / 24) * lambda * (lambda + 1) * (lambda + 2) * M4.w
  out <- -term1 + term2 - term3
  out
}
```

Define a custom moment function

The default function for `momentFUN` is `set.portfolio.moments`. We need to write our own function to estimate the moments for our objective function.

```r
crra.moments <- function(R, ...) {
    out <- list()
    out$mu <- colMeans(R)
    out$sigma <- cov(R)
    out$m3 <- PerformanceAnalytics:::M3.MM(R)
    out$m4 <- PerformanceAnalytics:::M4.MM(R)
    out
}
```

Specify Portfolio

```r
# Specify portfolio
portf.crra <- portfolio.spec(funds)

# Add constraint such that the weights sum to 1
portf.crra <- add.constraint(portf.crra, type="weight_sum",
                             min_sum=0.99, max_sum=1.01)

# Add box constraint such that no asset can have a weight of greater than
# 40% or less than 5%
portf.crra <- add.constraint(portf.crra, type="box",
                             min=0.05, max=0.4)

# Add objective to maximize CRRA
portf.crra <- add.objective(portf.crra, type="return",
                            name="CRRA", arguments=list(lambda=10))
```

"Dummy" Objectives

```
# Dummy objectives for plotting and/or further analysis
portf.crra <- add.objective(portf.crra, type="return", name="mean", multiplier=0)
portf.crra <- add.objective(portf.crra, type="risk", name="ES", multiplier=0)
portf.crra <- add.objective(portf.crra, type="risk", name="StdDev", multiplier=0)
```

Run Optimization

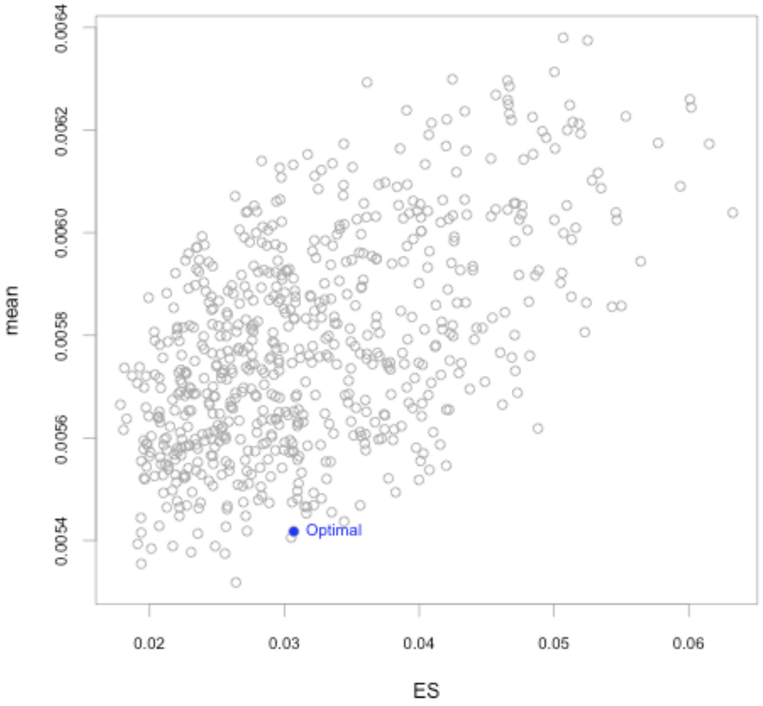
```
opt.crra <- optimize.portfolio(R, portf.crra, optimize_method="DEoptim",
                                 search_size=5000, trace=TRUE, traceDE=0,
                                 momentFUN="crra.moments")
```

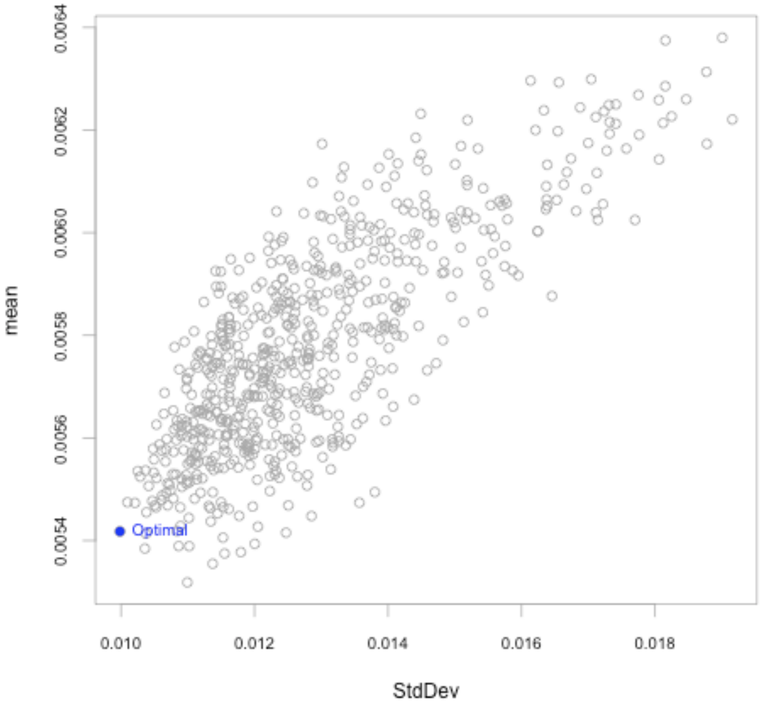
```
head(extractStats(opt.crra),4)
```

```
##              CRRA.CRRA     mean      ES  StdDev       out   w.CA  w.EMN
## .DE.portf.1 -0.0009786 0.005906 0.03407 0.01367 0.0009786 0.1667 0.1667
## .DE.portf.2 -0.0007585 0.005716 0.02532 0.01215 0.0007585 0.1560 0.3420
## .DE.portf.3 -0.0019451 0.006380 0.05068 0.01901 0.0019451 0.1180 0.1200
## .DE.portf.4 -0.0007816 0.005536 0.02591 0.01232 0.0007816 0.0900 0.1840
##              w.FIA w.CTAG   w.EM   w.GM
## .DE.portf.1 0.1667 0.1667 0.1667 0.1667
## .DE.portf.2 0.0560 0.2180 0.1280 0.0920
## .DE.portf.3 0.0880 0.1300 0.3900 0.1620
## .DE.portf.4 0.2800 0.2720 0.1160 0.0640
```

Chart Results

```
chart.RiskReward(opt.crra, risk.col = "ES")
chart.RiskReward(opt.crra, risk.col = "StdDev")
```





Run Backtest and Compute Returns

```
bt.opt.crra <- optimize.portfolio.rebalancing(R, portf.crra,
                                              optimize_method="DEoptim",
                                              search_size=5000, trace=TRUE,
                                              traceDE=0,
                                              momentFUN="crra.moments",
                                              rebalance_on=rebal.freq,
                                              training_period=training,
                                              trailing_periods=trailing)

ret.crra <- summary(bt.opt.crra)$portfolio_returns
colnames(ret.crra) <- "CRRA"
```

Chart Weights Through Time

```
chart.Weights(bt.opt.crra, main="CRRA Weights", col=bluemono)
```

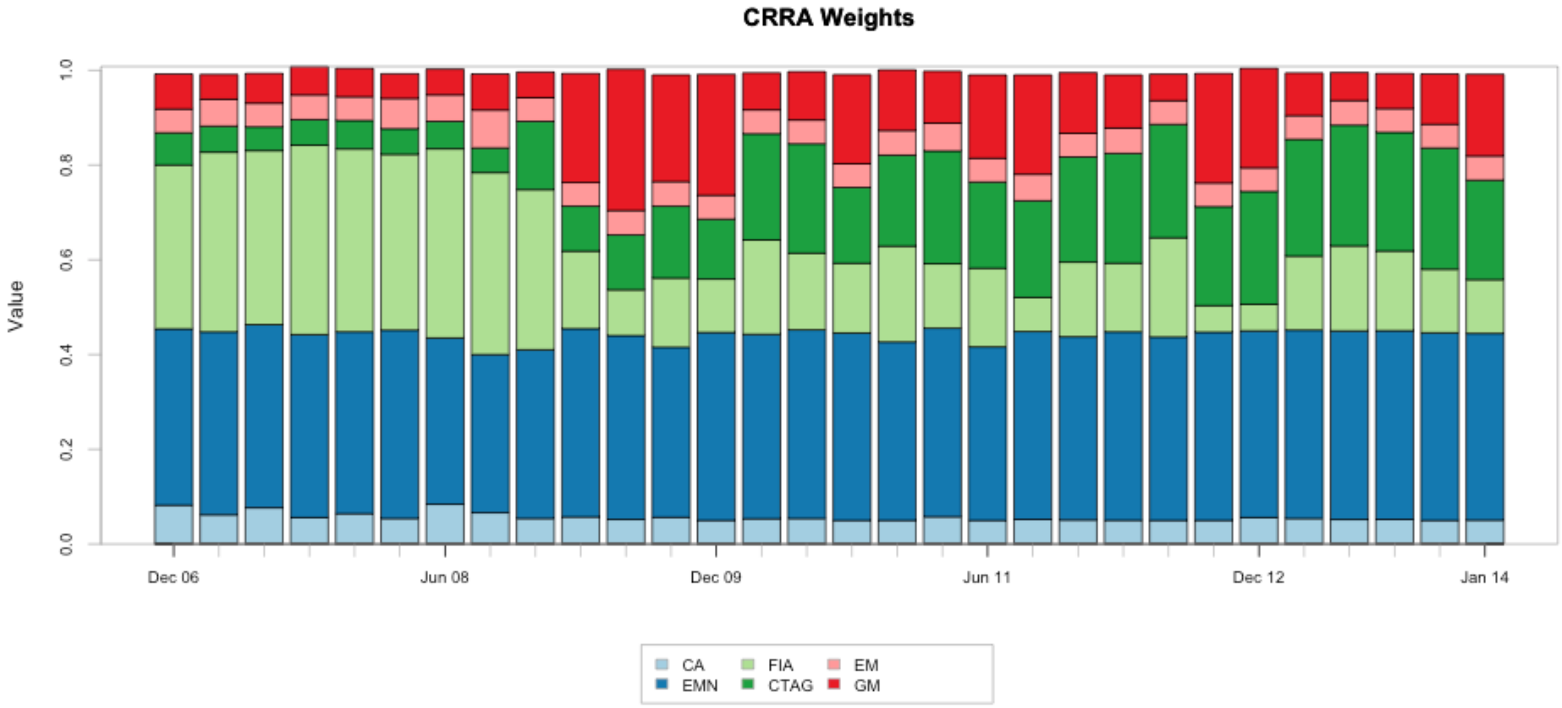
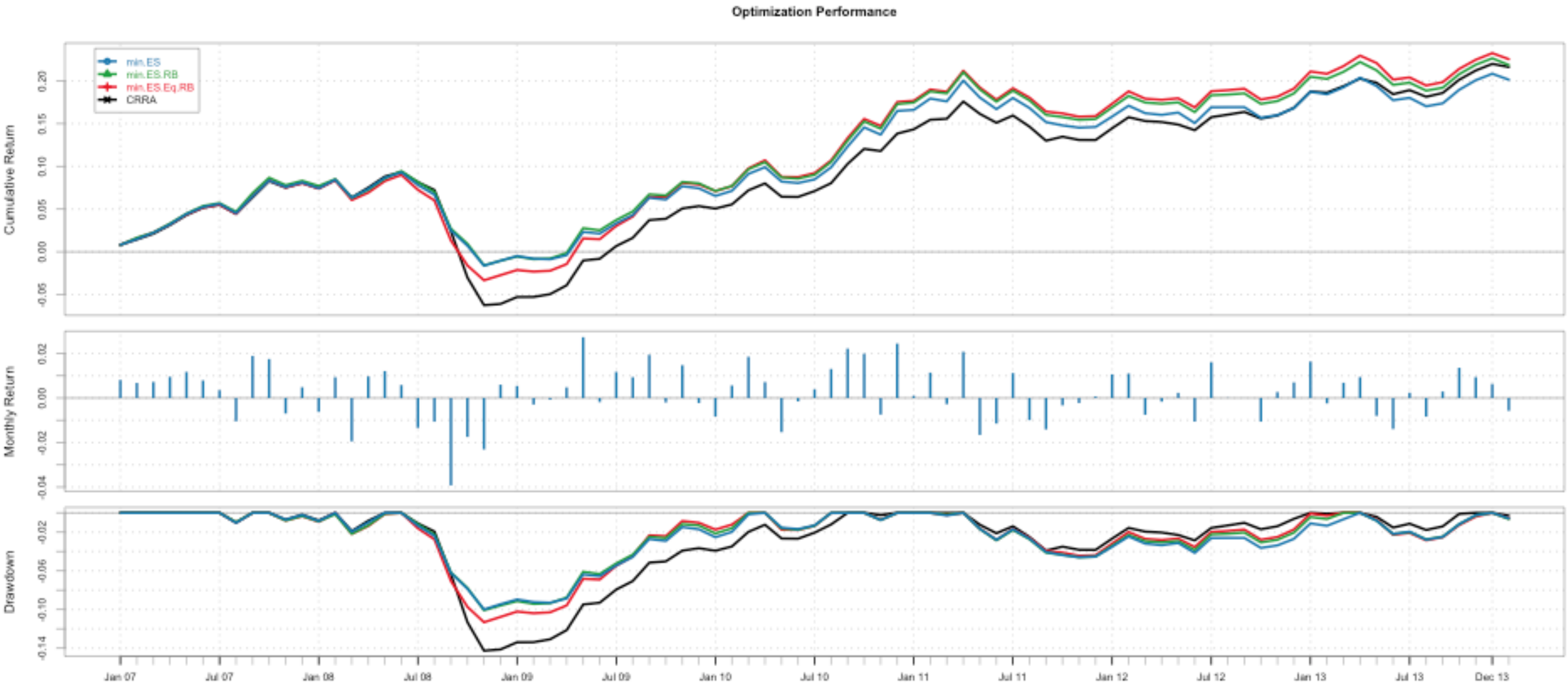


Chart Performance

```
charts.PerformanceSummary(cbind(ret.bt.opt, ret.crra),
                                            main="Optimization Performance")
```



Conclusion

- Introduced the goals and summary of PortfolioAnalytics
- Demonstrated the flexibility through examples
- Exciting plans for GSOC 2014
  - Support for regime switching
  - Support for supervised learning
  - many more

Acknowledgements

Many thanks to...

- Google: funding for Google Summer of Code (GSoC)
- UW CF&RM Program: continued work on PortfolioAnalytics
- GSoC Mentors: Brian Peterson, Peter Carl, Doug Martin, and Guy Yollin
- R/Finance Committee

PortfolioAnalytics Links

PortfolioAnalytics is on R-Forge in the ReturnAnalytics project

- PortfolioAnalytics

Source code for the slides

- https://github.com/rossb34/PortfolioAnalyticsPresentation

and view it here

- http://rossb34.github.io/PortfolioAnalyticsPresentation/

Any Questions?

References and Useful Links

- ROI
- DEoptim
- pso
- GenSA
- PerformanceAnalytics
- Patrick Burns Random Portfolios
- W.T. Shaw Random Portfolios
- Martellini paper
- Boudt paper
- Shiny App