R与投资组合分析

邓一硕
http://yishuo.org
COS金融量化分析版
2011年11月13日

引子

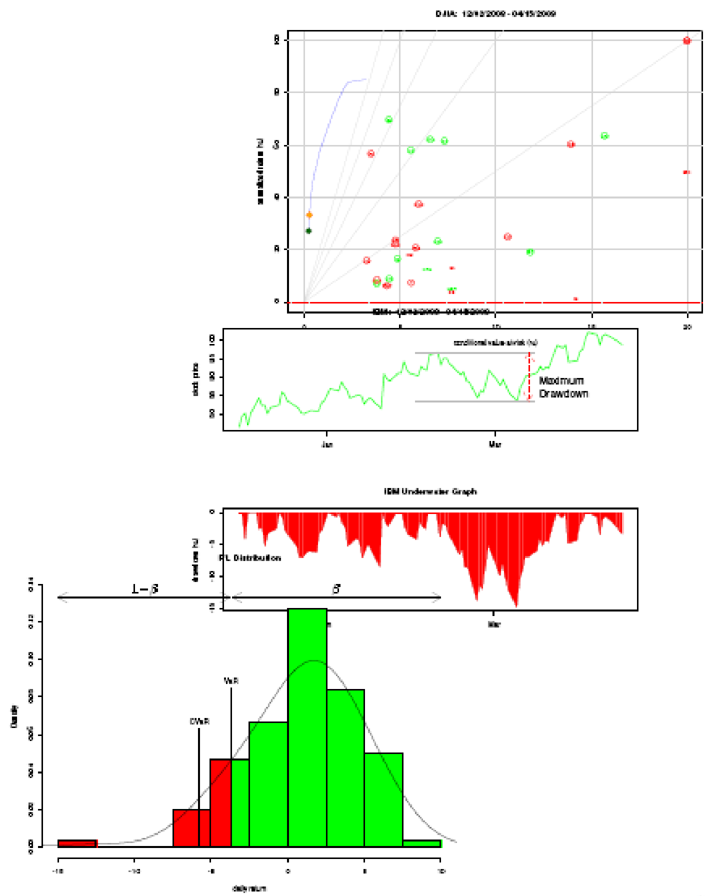
- **R-SIG-FINANCE QUESTION:**

  Can I do < fill in the blank >

  portfolio optimization in R?

- **ANSWER:**

  Yes! (98% confidence level)



大纲

- 基于均值-方差的投资组合优化
- 基于**CVaR**的投资组合优化
- 其它投资组合优化方法
  - 差分优化算法
  - **Omega**
  - **Max DrawDown**
  - Rachev Ratio

基于MV的投资组合优化

- Markowitz投资组合理论
- $$Min: w^T\Omega w$$
  $$St: \sum_i \bar{r}_i w_i = r_{min}$$
  $$\sum_i w_i = 1$$
  $$w_i^{min} \le w_i \le w_i^{max}$$
- 函数： portfolio.optim()
- 用法：

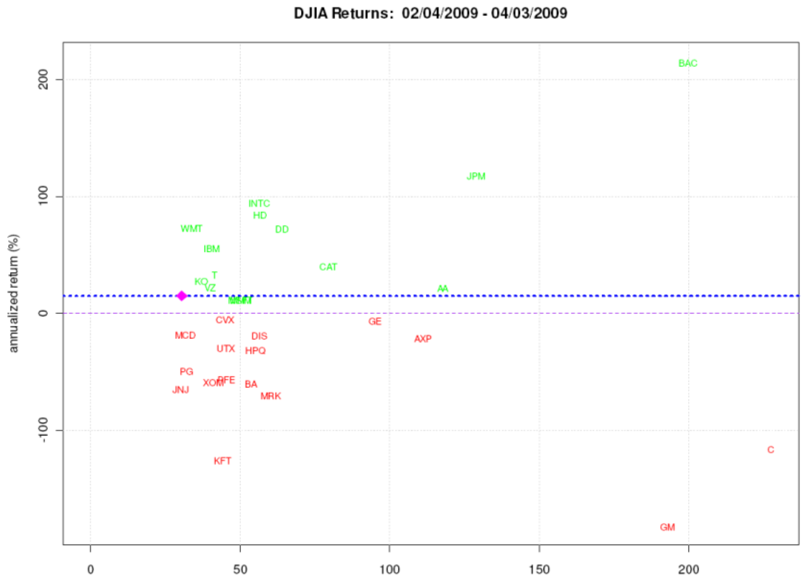
  portfolio.optim(x,pm=mean(x),riskless=FALSE,shorts=FALSE,rf=0,reslow=NULL,reshigh=NULL,covmat=cov(x),...)

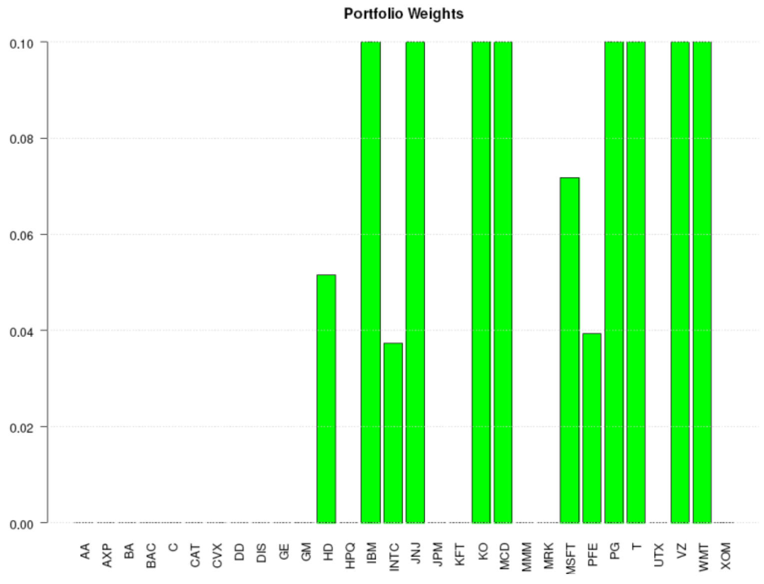
基于MV的投资组合优化

- 例子：

```
> averet=matrix(colMeans(r),nrow=1)
> rcov=cov(r)
> target.return = 15/250
> port.sol=portfolio.optim(x=averet,pm=target.return,covmat=rcov,shorts=F,
  reslow=rep(0,30),reshigh=rep(0.1,30))
> w=round(port.sol$pw,3)
> colnames(w)=colnames(r)
> w[w>=10^(-17)]
 [1] 0.100 0.100 0.100 0.074 0.100 0.100 0.100 0.094 0.032 0.007 0.094 0.100
```

组合优化结果

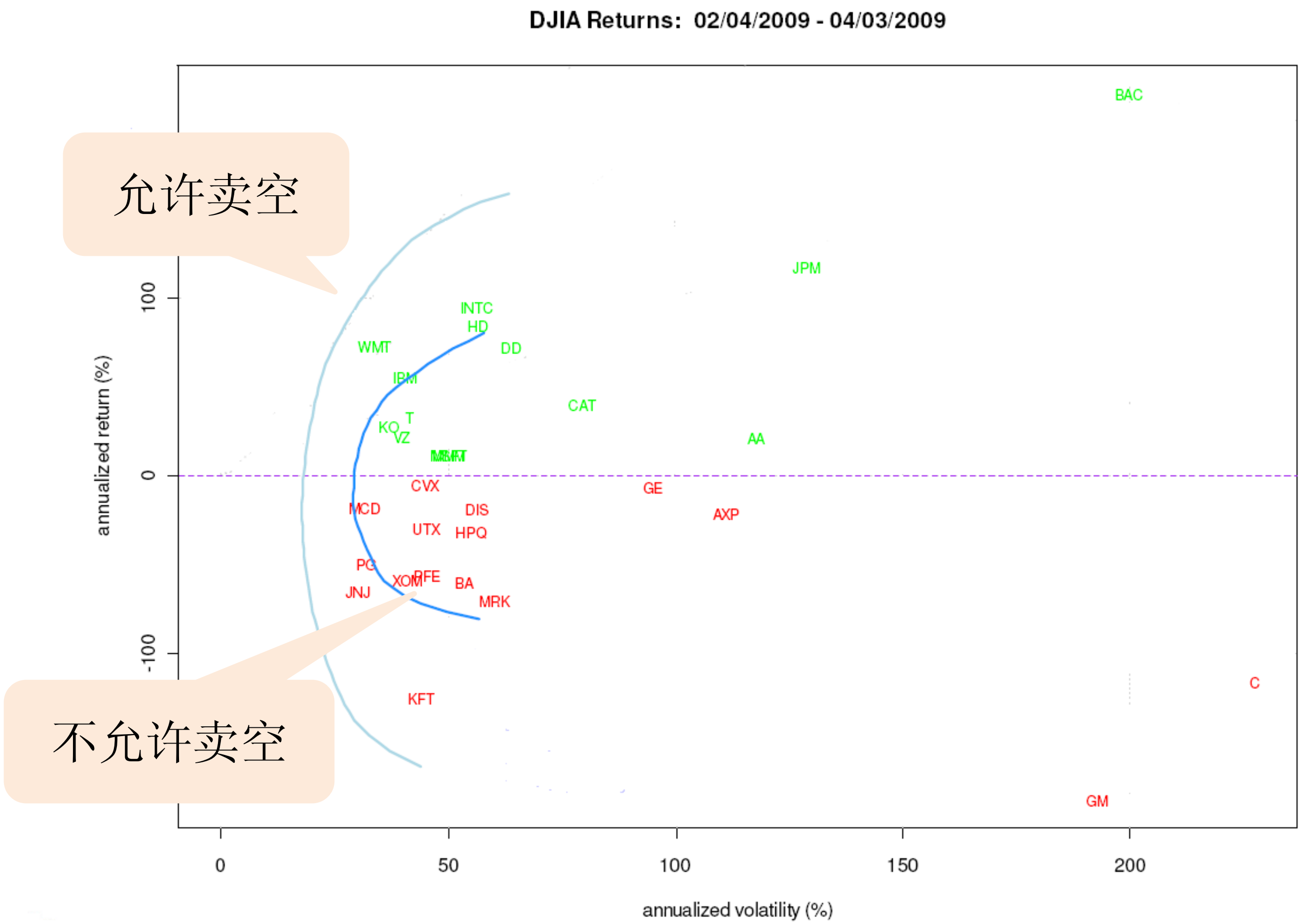

DJIA Returns: 02/04/2009 - 04/03/2009
annualized return (%)
BAC
JPM
INTC
HD
WMT
DD
IBM
CAT
T
KO
VZ
AA
CVX
GE
MCD
DIS
AXP
UTX
HPQ
PG
PFE
XOM
BA
JNJ
MRK
KFT
C
GM


Portfolio Weights
AA AXP BA BAC C CAT CVX DD DIS GE GM HD HPQ IBM INTC JNJ JPM KFT KO MCD MMM MRK MSFT PFE PG T UTX VZ WMT XOM

effFrontier函数

```
effFrontier = function (averet, rcov, nports = 20, shorts=T, wmax=1)
{
mxret = max(abs(averet))
mnret = -mxret
n.assets = ncol(averet)
reshigh = rep(wmax,n.assets)
if( shorts )
{
reslow = rep(-wmax,n.assets)
} else {
reslow = rep(0,n.assets)
}
min.rets = seq(mnret, mxret, len = nports)
vol = rep(NA, nports)
ret = rep(NA, nports)
for (k in 1:nports)
{
port.sol = NULL
try(port.sol <- portfolio.optim(x=averet, pm=min.rets[k], covmat=rcov,
reshigh=reshigh, reslow=reslow,shorts=shorts),silent=T)
if ( !is.null(port.sol) )
{
vol[k] = sqrt(as.vector(port.sol$pw %*% rcov %*% port.sol$pw))
ret[k] = averet %*% port.sol$pw
}
}
return(list(vol = vol, ret = ret))
}
```

有效边界

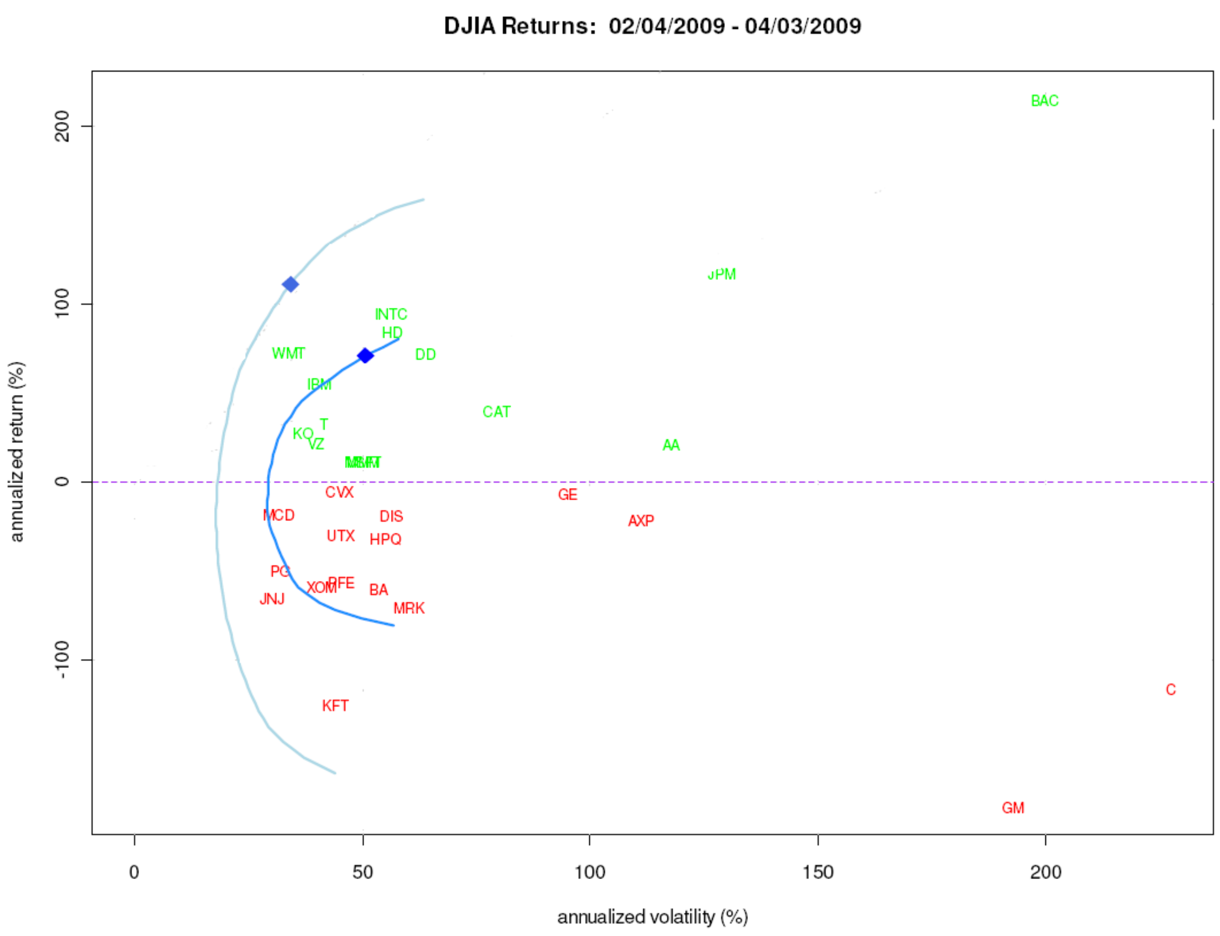


DJIA Returns: 02/04/2009 - 04/03/2009
允许卖空
不允许卖空
annualized return (%)
annualized volatility (%)
100
0
-100
0
50
100
150
200
BAC
JPM
INTC
HD
DD
WMT
IBM
CAT
T
KO
VZ
AA
CVX
GE
MCD
DIS
AXP
UTX
HPQ
PG
PFE
XOM
BA
JNJ
MRK
KFT
C
GM

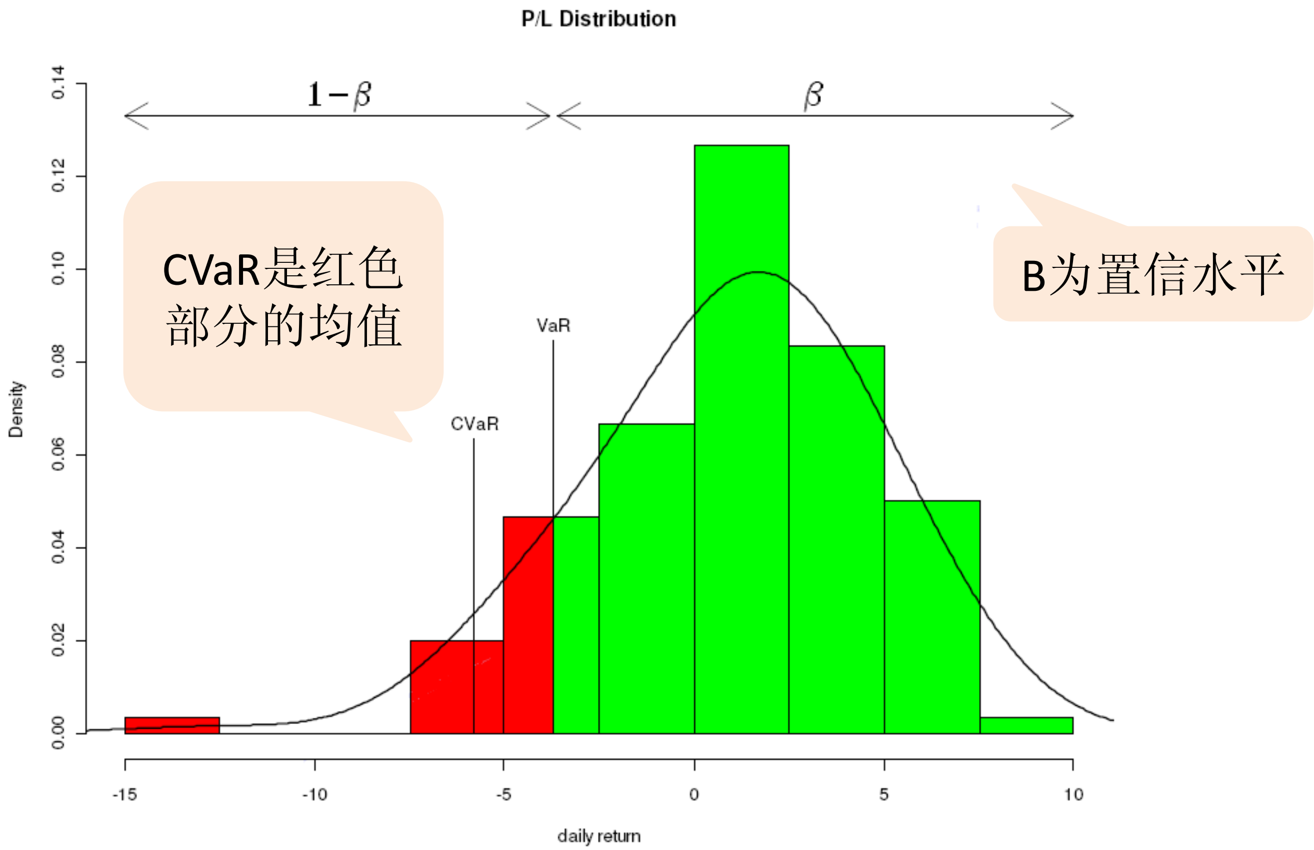
Sharpe Ratio

```
maxSharpe = function (averet, rcov, shorts=T, wmax = 1)
{
optim.callback = function(param,averet,rcov,reshigh,reslow,shorts)
{
port.sol = NULL
try(port.sol <- portfolio.optim(x=averet, pm=param, covmat=rcov,
reshigh=reshigh, reslow=reslow, shorts=shorts), silent = T)
if (is.null(port.sol)) {
ratio = 10^9
} else {
m.return = averet %*% port.sol$pw
m.risk = sqrt(as.vector(port.sol$pw %*% rcov %*% port.sol$pw))
ratio = -m.return/m.risk
assign("w",port.sol$pw,inherits=T)
}
return(ratio)
}
ef = effFrontier(averet=averet, rcov=rcov, shorts=shorts, wmax=wmax, nports = 100)
n = ncol(averet)
reshigh = rep(wmax,n)
if( shorts ) {
reslow = -reshigh
} else {
reslow = rep(0,n)
}
max.sh = which.max(ef$ret/ef$vol)
w = rep(0,ncol(averet))
xmin = optimize(f=optim.callback, interval=c(ef$ret[max.sh-1], upper=ef$ret[max.sh+1]),
averet=averet,rcov=rcov,reshigh=reshigh,reslow=reslow,shorts=shorts)
return(w)
}
```

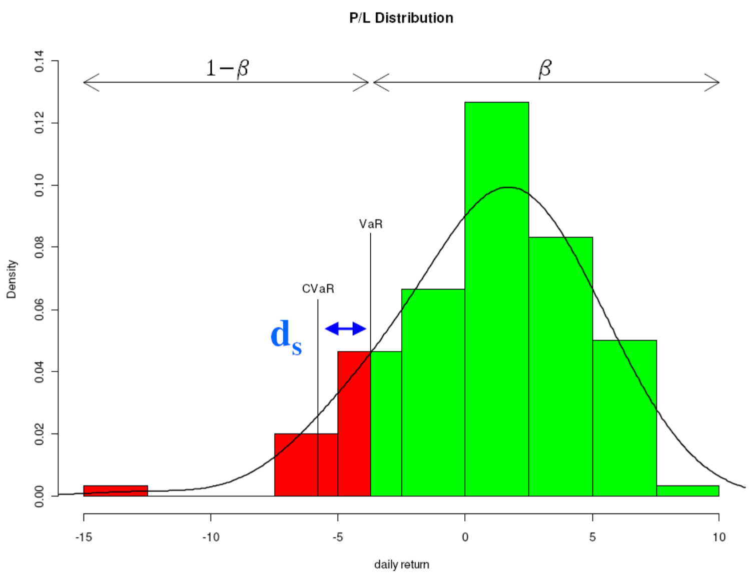
最大shape比率


DJIA Returns: 02/04/2009 - 04/03/2009
annualized return (%)
annualized volatility (%)
200
100
0
-100
0
50
100
150
200
BAC
JPM
INTC
HD
DD
WMT
IBM
CAT
T
KO
VZ
AA
CVX
GE
MCD
DIS
AXP
UTX
HPQ
PG
PFE
XOM
BA
JNJ
MRK
KFT
C
GM

基于CVaR的投资组合优化



基于CVaR的投资组合优化



- CVaR

$$R_{CVaR}(w,\beta) = R_{VaR} + \frac{\frac{1}{S}\sum_1^S max(R_{VaR} - w'R_s, 0)}{1-\beta}$$

- CVaR优化

$$Min: R_{VaR} + \frac{1}{S}\frac{1}{1-\beta}\sum^S d_s$$
$$S.t: d_s \geq R_{VaR} + w'\vec{R}_s$$
$$w'\bar{R} \geq R_{min}$$
$$\sum_i W_i = 1$$

参考：B. Sherer 2003

一般线性规划

- 函数： Rglpk_solve_LP
- 两者的关系：

一般线性规划

CVaR投资组合优化

$$\text{Minimize:} \quad c^T x$$
$$\text{Subject to:} \quad Ax \geq b_0$$

$$\Longrightarrow$$

$$\mathbf{c^T} = \begin{bmatrix} 0 & 0 & \dots & 0 & \frac{-1}{(1-\beta)S} & \frac{-1}{(1-\beta)S} & \cdots & \frac{-1}{(1-\beta)S} & -1 \end{bmatrix}$$

$$\mathbf{x^T} = \begin{bmatrix} w_1 & w_2 & \dots & w_n & d_1 & d_2 & \dots & d_S & R_{VaR} \end{bmatrix}$$

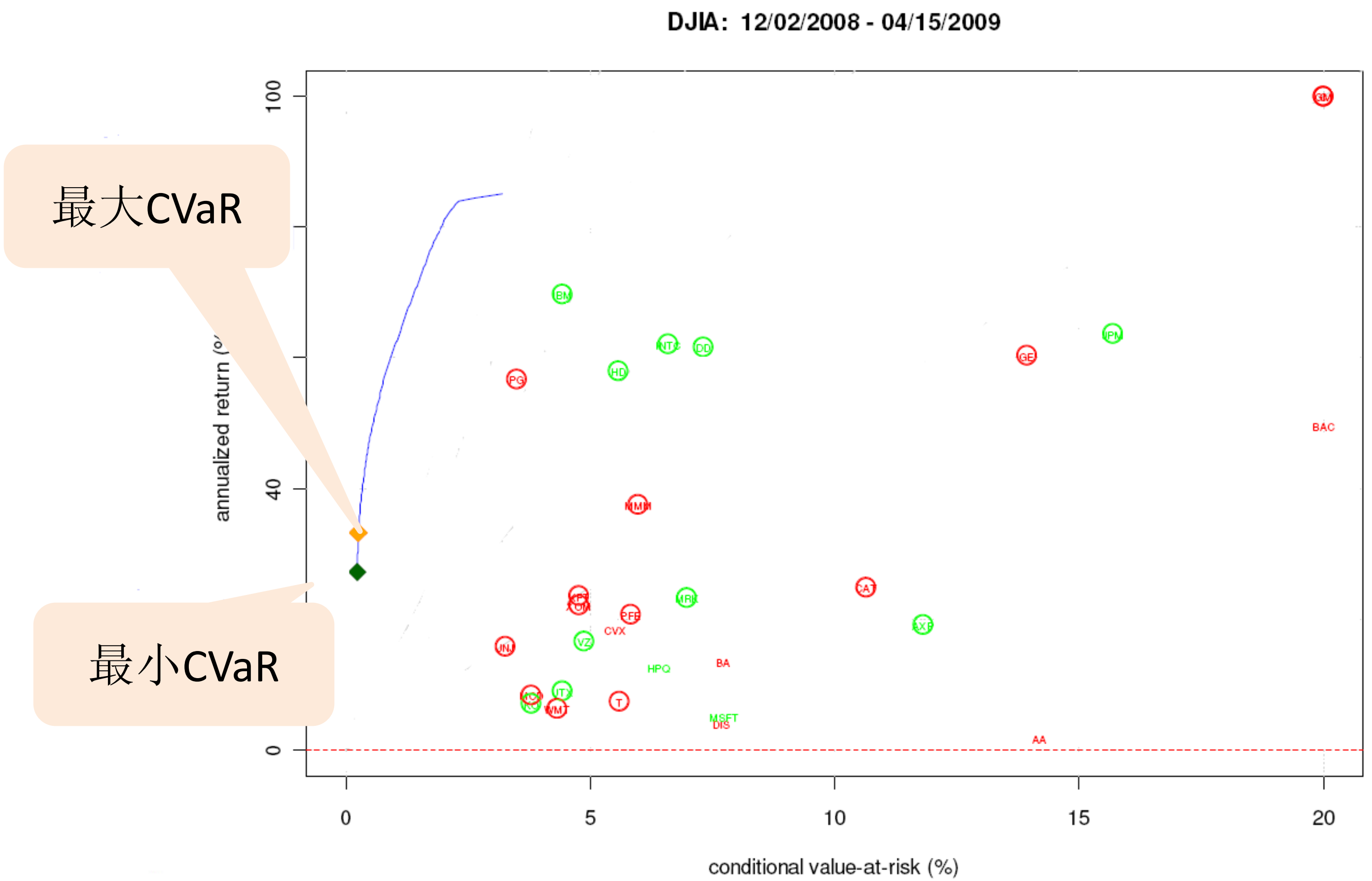
$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 & \dots & 0 & 0 \\ r_1 & r_2 & \dots & r_n & 0 & \dots & 0 & 0 \\ r_{11} & r_{12} & \dots & r_{1n} & 1 & 0 & \dots & 1 \\ r_{21} & r_{22} & \dots & r_{2n} & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 1 \\ r_{s1} & r_{s2} & \dots & r_{sn} & 0 & \dots & 1 & 1 \end{bmatrix} \qquad \mathbf{b_0} = \begin{bmatrix} 1 \\ rmin \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

CVaR优化函数

```
cvarOpt = function(rmat, alpha=0.05, rmin=0, wmin=0, wmax=1, weight.sum=1)
{
require(Rglpk)
n = ncol(rmat) # number of assets
s = nrow(rmat) # number of scenarios i.e. periods
averet = colMeans(rmat)
# creat objective vector, constraint matrix, constraint rhs
Amat = rbind(cbind(rbind(1,averet),matrix(data=0,nrow=2,ncol=s+1)),
cbind(rmat,diag(s),1))
objL = c(rep(0,n), rep(-1/(alpha*s), s), -1)
bvec = c(weight.sum,rmin,rep(0,s))
# direction vector
dir.vec = c("==",">=",rep(">=",s))
# bounds on weights
bounds = list(lower = list(ind = 1:n, val = rep(wmin,n)),
upper = list(ind = 1:n, val = rep(wmax,n)))
res = Rglpk_solve_LP(obj=objL, mat=Amat, dir=dir.vec, rhs=bvec,
types=rep("C",length(objL)), max=T, bounds=bounds)
w = as.numeric(res$solution[1:n])
return(list(w=w,status=res$status))
}
```

支持不等号

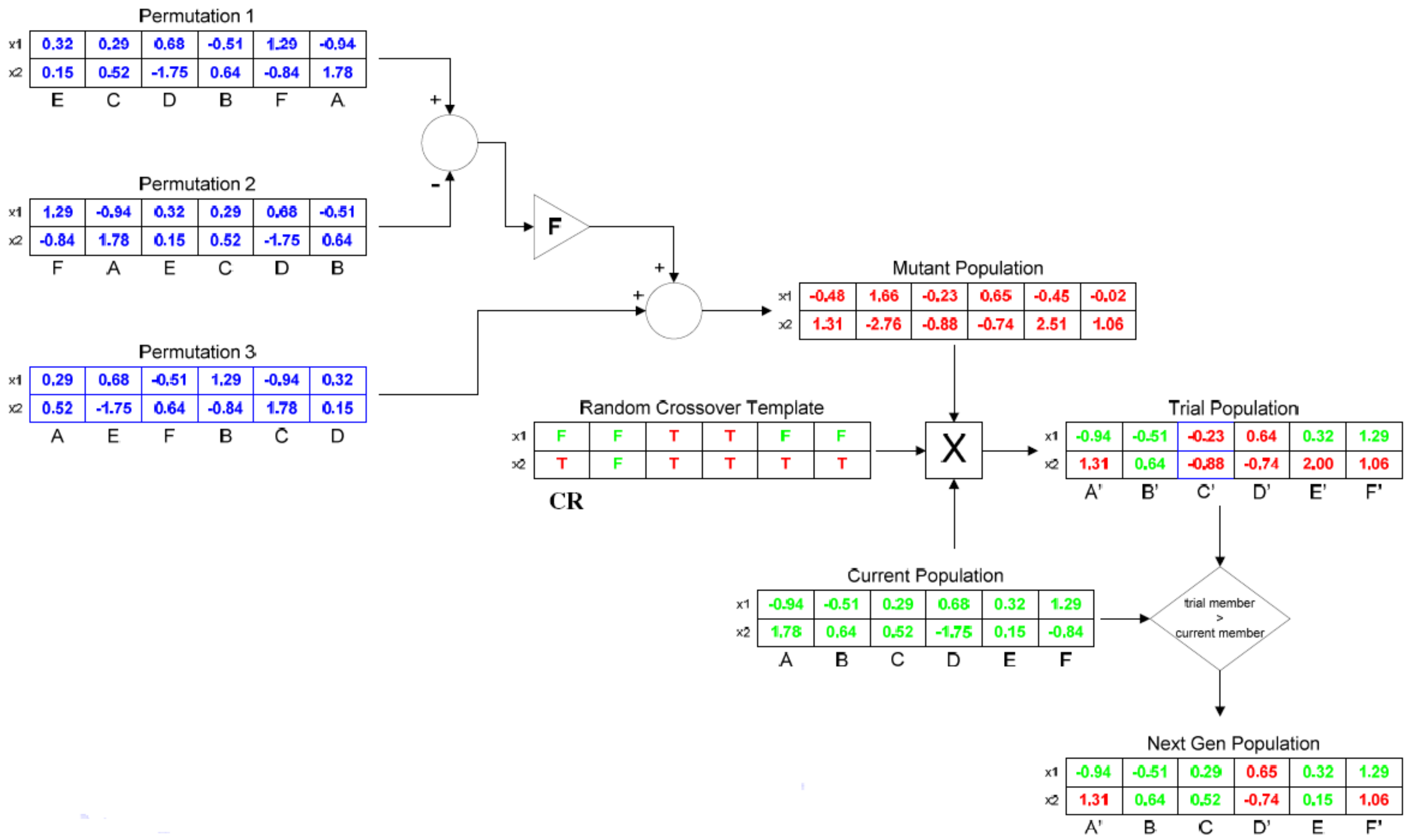
CVaR有效边界


DJIA: 12/02/2008 - 04/15/2009
annualized return (%)
conditional value-at-risk (%)
0
40
100
0
5
10
15
20
BAC
CVX
HPQ
BA
MSFT
DIS
AA
最大CVaR
最小CVaR

差分进化算法简介

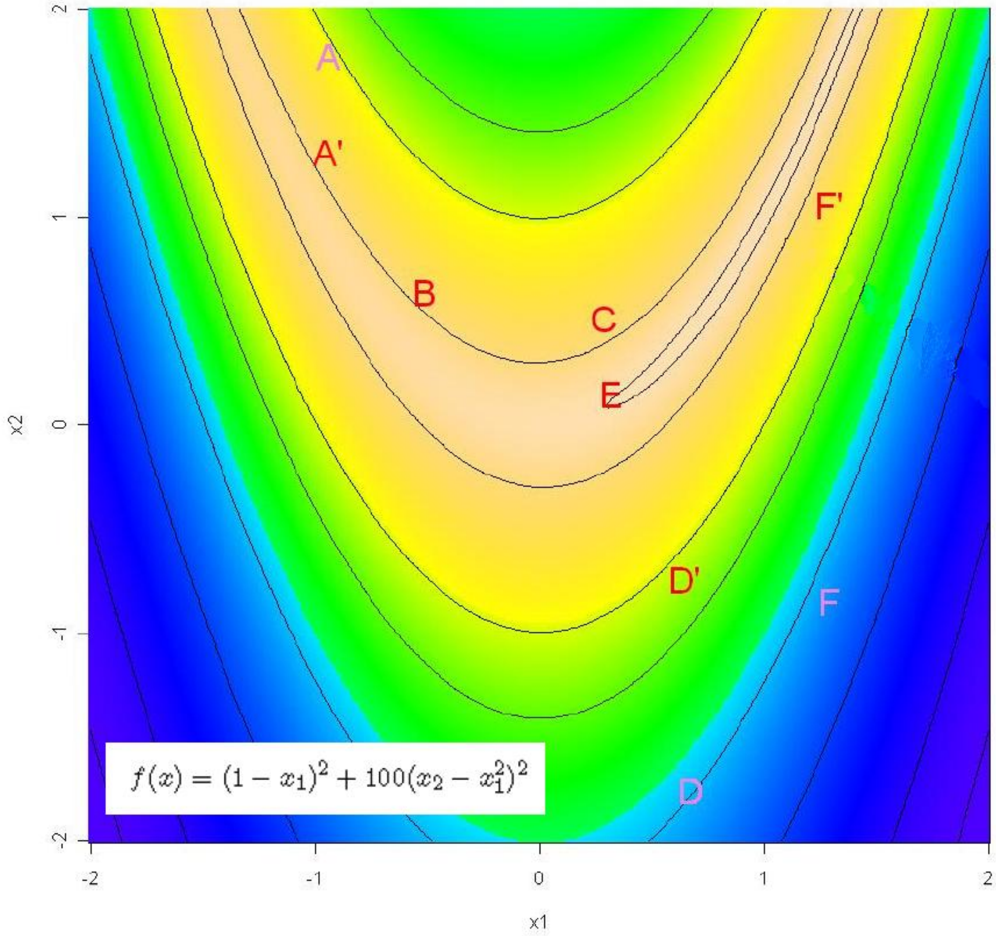
- 1990，Pricehe&Storn提出
- 原理简单，功能强大
- R包：
  - DEoptim

差分进化算法简介



Permutation 1
x1 0.32 0.29 0.68 -0.51 1.29 -0.94
x2 0.15 0.52 -1.75 0.64 -0.84 1.78
E C D B F A
Permutation 2
x1 1.29 -0.94 0.32 0.29 0.68 -0.51
x2 -0.84 1.78 0.15 0.52 -1.75 0.64
F A E C D B
Permutation 3
x1 0.29 0.68 -0.51 1.29 -0.94 0.32
x2 0.52 -1.75 0.64 -0.84 1.78 0.15
A E F B C D
F
Mutant Population
x1 -0.48 1.66 -0.23 0.65 -0.45 -0.02
x2 1.31 -2.76 -0.88 -0.74 2.51 1.06
Random Crossover Template
x1 F F T T F F
x2 T F T T T T
CR
X
Trial Population
x1 -0.94 -0.51 -0.23 0.64 0.32 1.29
x2 1.31 0.64 -0.88 -0.74 2.00 1.06
A' B' C' D' E' F'
Current Population
x1 -0.94 -0.51 0.29 0.68 0.32 1.29
x2 1.78 0.64 0.52 -1.75 0.15 -0.84
A B C D E F
trial member > current member
Next Gen Population
x1 -0.94 -0.51 0.29 0.65 0.32 1.29
x2 1.31 0.64 0.52 -0.74 0.15 1.06
A' B C D' E F'

DE示例

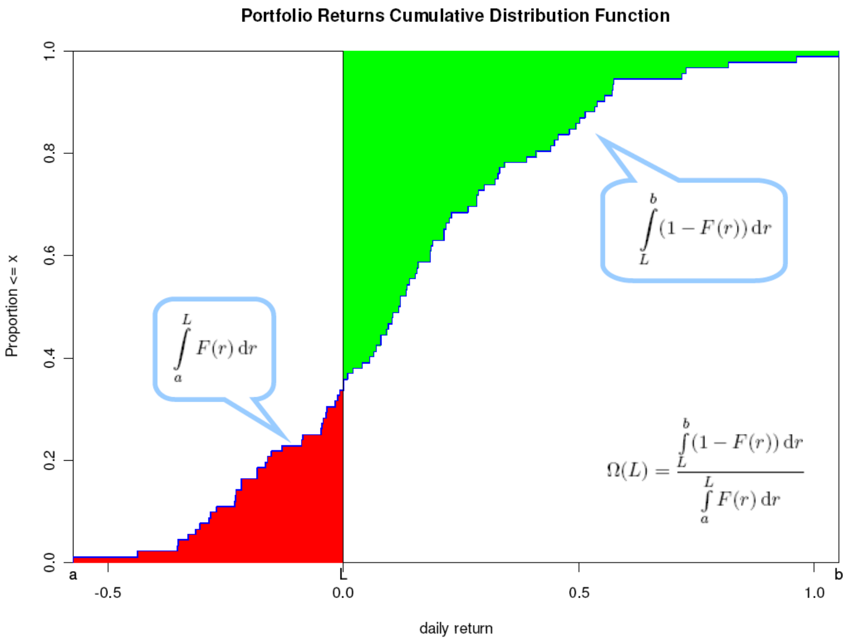


差分进化算法函数

- 函数
  - Deoptim
- 用法
  - DEoptim(FUN,lower,upper,coontrol=list(),…)
- 例子:

lower=c(-2,-2)
upper=c(2,2)
Res=DEoptim(f,lower,upper)
Res$optim
$bestmem
par1 par2
0.9987438 0.9976079
$bestval
[1] 2.986743e-06
$nfeval
[1] 5050
$iter
[1] 100

Omega投资组合优化



- Omega：

$$\Omega(L) = \frac{\int_L^b (1-F(r))dr}{\int_a^L F(r)dr}$$

- call price/put price

$$\Omega(L) = \frac{C(L)}{P(L)}$$

- L:Strike price
- 简单算法
- Omega=mean(pmax(r-l,0))/mean(pmax(L-r,0))

参考：Kazemi et. al., 2003
Keating & Shadwick 2002

Omega优化函数

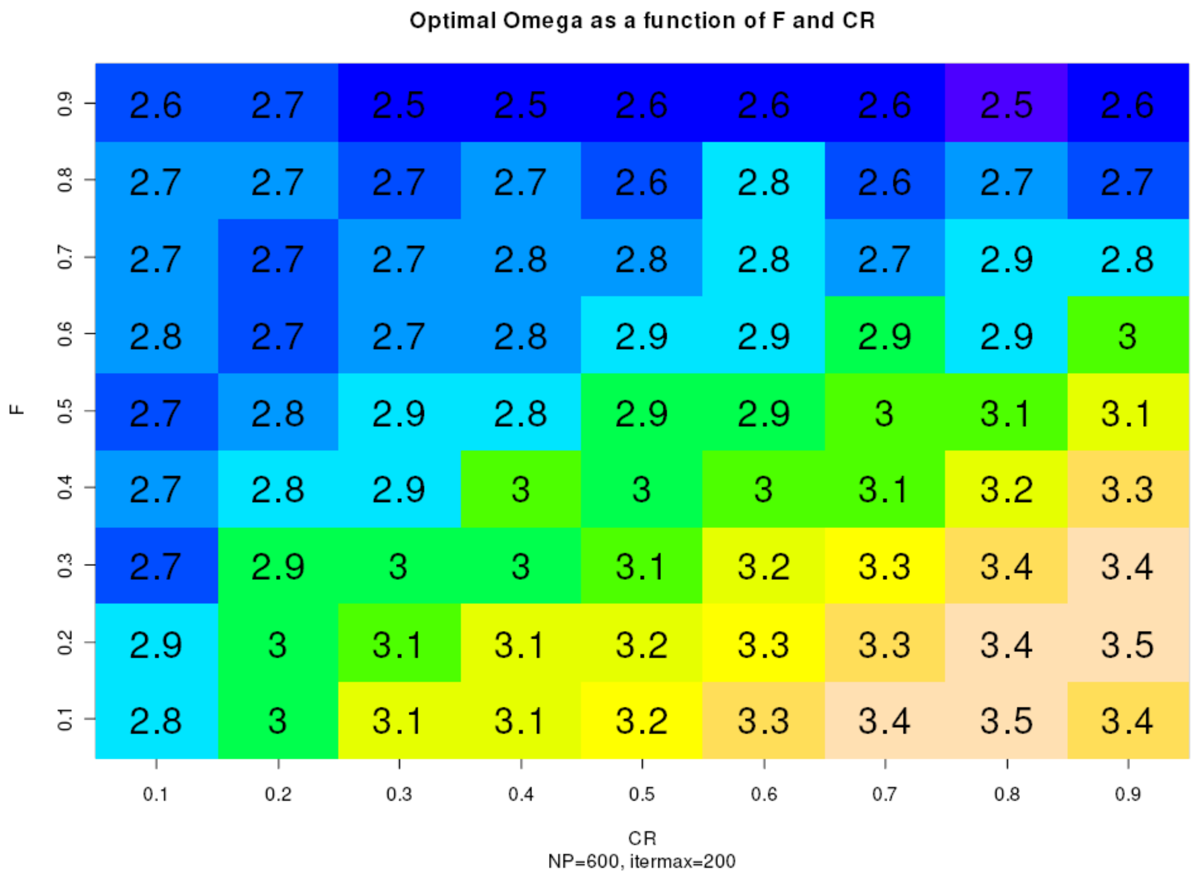
```
optOmega = function(x,ret,L)
{
retu = ret %*% x
obj = -Omega(retu,L=L,method="simple")
weight.penalty = 100*(1-sum(x))^2
return( obj + weight.penalty )
}
> lower = rep(0,n.assets)
> upper = rep(wmax,n.assets)
> res = DEoptim(optOmega,lower,upper,
control=list(NP=2000,itermax=1000,F=0.2,CR=0.8),
ret=coredata(r),L=L)
> w = cleanWeights(res$optim$bestmem,syms)
> w[w!=0]
AXP BA C CAT CVX DD DIS GE GM HD IBM INTC JNJ KO MCD MMM
0.02 0.03 0.02 0.04 0.05 0.08 0.01 0.02 0.01 0.03 0.04 0.09 0.05 0.08 0.05 0.04
MRK PG T UTX VZ WMT XOM
0.04 0.10 0.08 0.04 0.06 0.03 0.00
```

Maximize: $\Omega(L)$

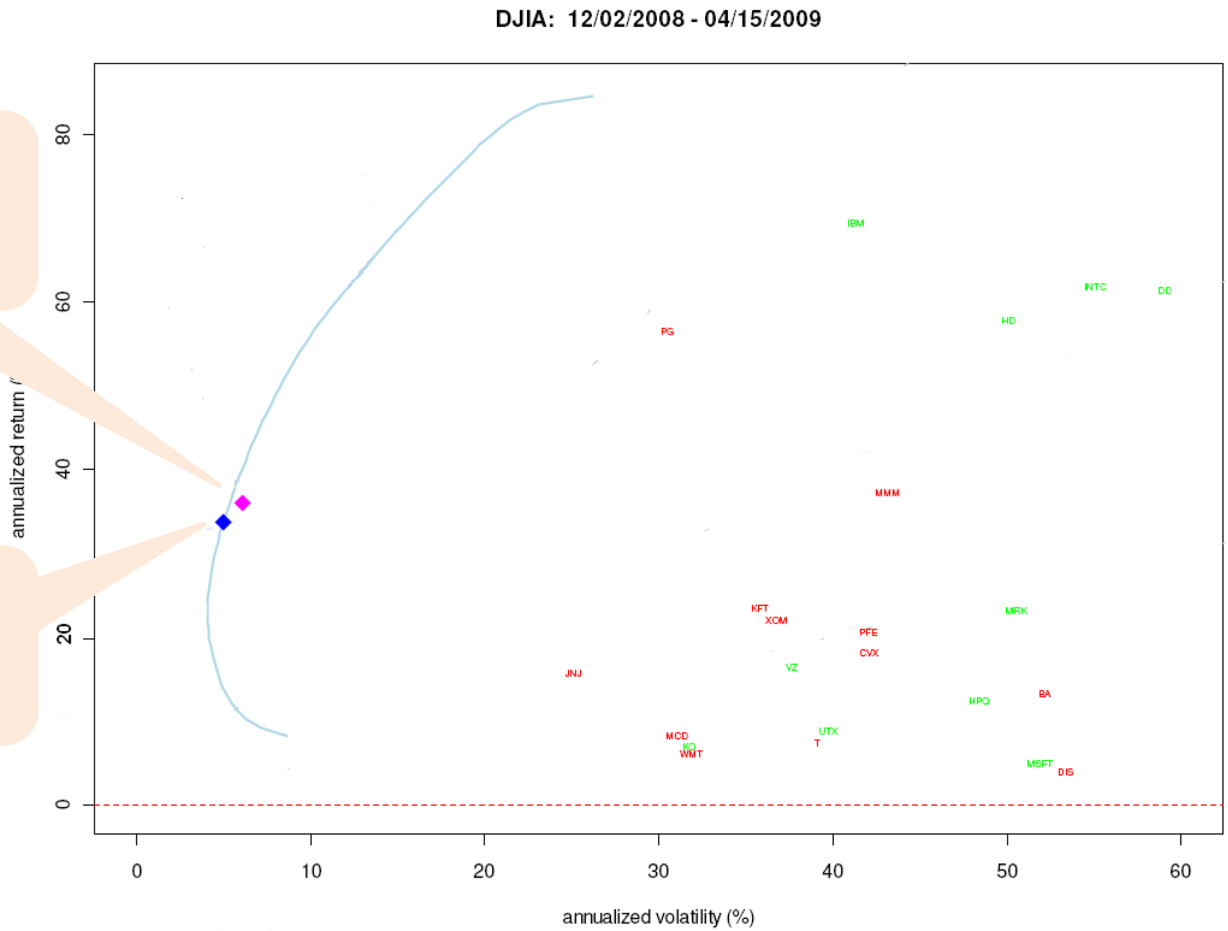
Subject to: $\sum_i |w_i| = 1$

$0 \leq w_i \leq w_i^{max}$

DE参数对Omega的影响



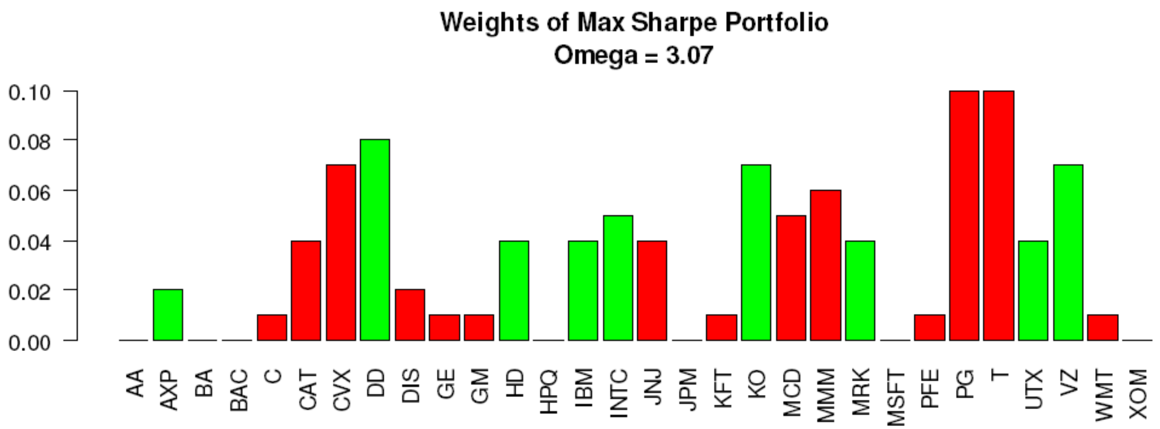
Max Omega VS Max Sharpe ratio

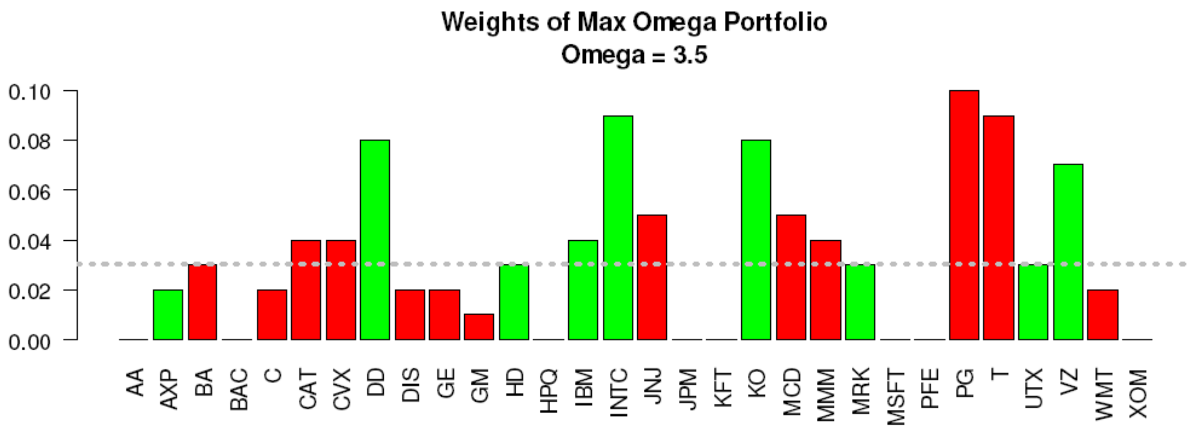


<u>最大Omega:</u>
Omega=3.50
Sharpe=5.92

<u>最大Sharpe:</u>
Sharpe=6.76
Omega=3.07

两种方法的权重对比

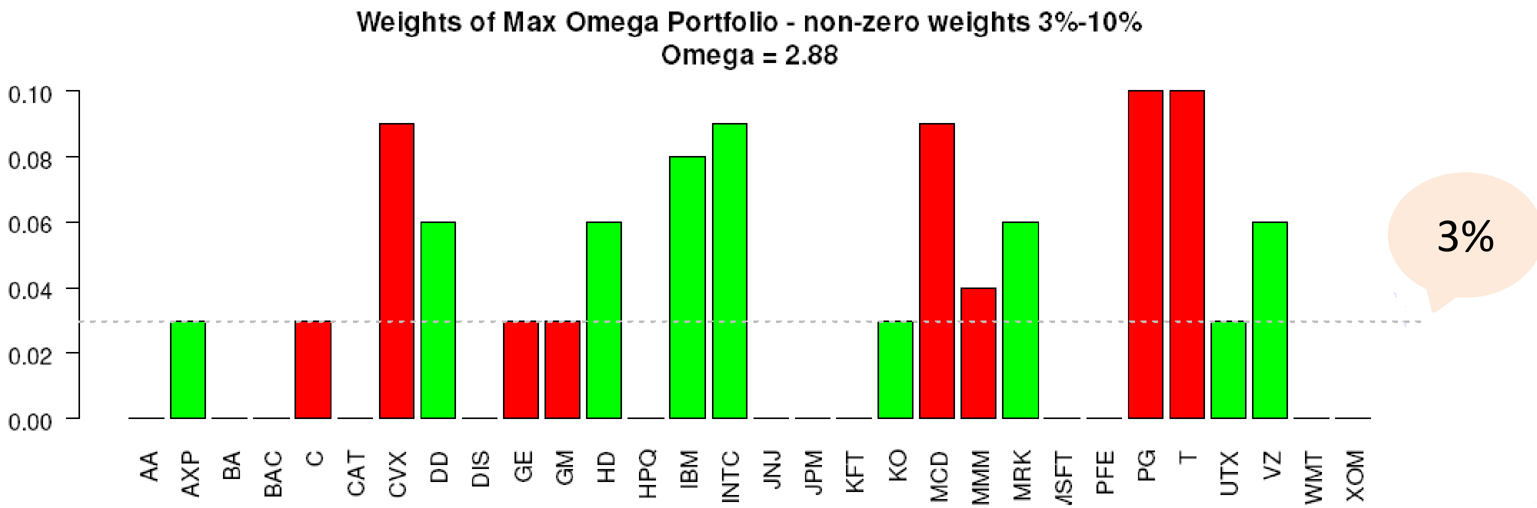

Weights of Max Sharpe Portfolio
Omega = 3.07
0.10
0.08
0.06
0.04
0.02
0.00
AA
AXP
BA
BAC
C
CAT
CVX
DD
DIS
GE
GM
HD
HPQ
IBM
INTC
JNJ
JPM
KFT
KO
MCD
MMM
MRK
MSFT
PFE
PG
T
UTX
VZ
WMT
XOM


Weights of Max Omega Portfolio
Omega = 3.5
0.10
0.08
0.06
0.04
0.02
0.00
AA
AXP
BA
BAC
C
CAT
CVX
DD
DIS
GE
GM
HD
HPQ
IBM
INTC
JNJ
JPM
KFT
KO
MCD
MMM
MRK
MSFT
PFE
PG
T
UTX
VZ
WMT
XOM

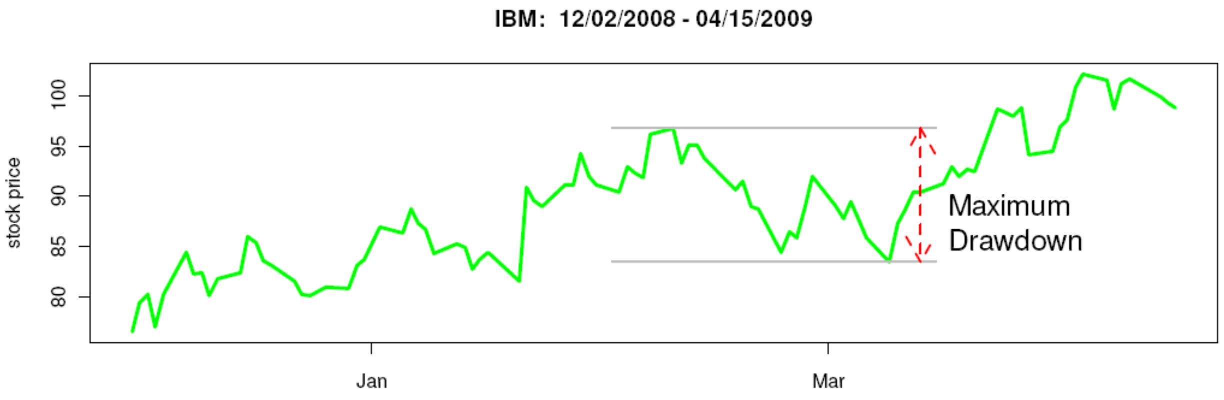
带约束的优化(weight[weight!=0]>.03)

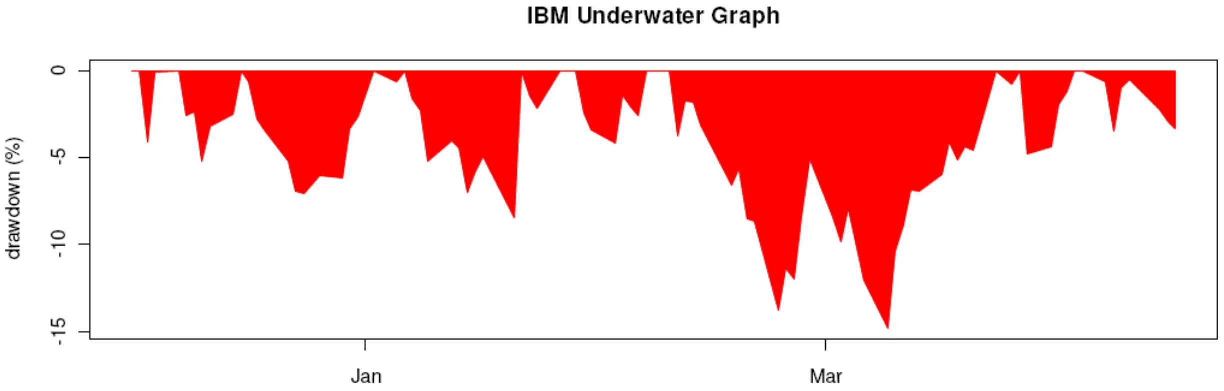
```
# max omega with non-zero weights between 3% & 10%
optOmega.gt3 = function(x,ret,L)
{
retu = ret %*% x
obj = -Omega(retu,L=L,method="simple")
weight.penalty = 100*(1-sum(x))^2
small.weight.penalty = 100*sum(x[x<0.03])
return( obj + weight.penalty + small.weight.penalty )
}
res = DEoptim(optOmega.gt3,lower,upper,
control=list(NP=2000,itermax=1000,F=0.2,CR=0.8),
ret=coredata(r),L=L)
```

带约束的优化(weight[weight!=0]>.03)



最大跌幅投资组合优化

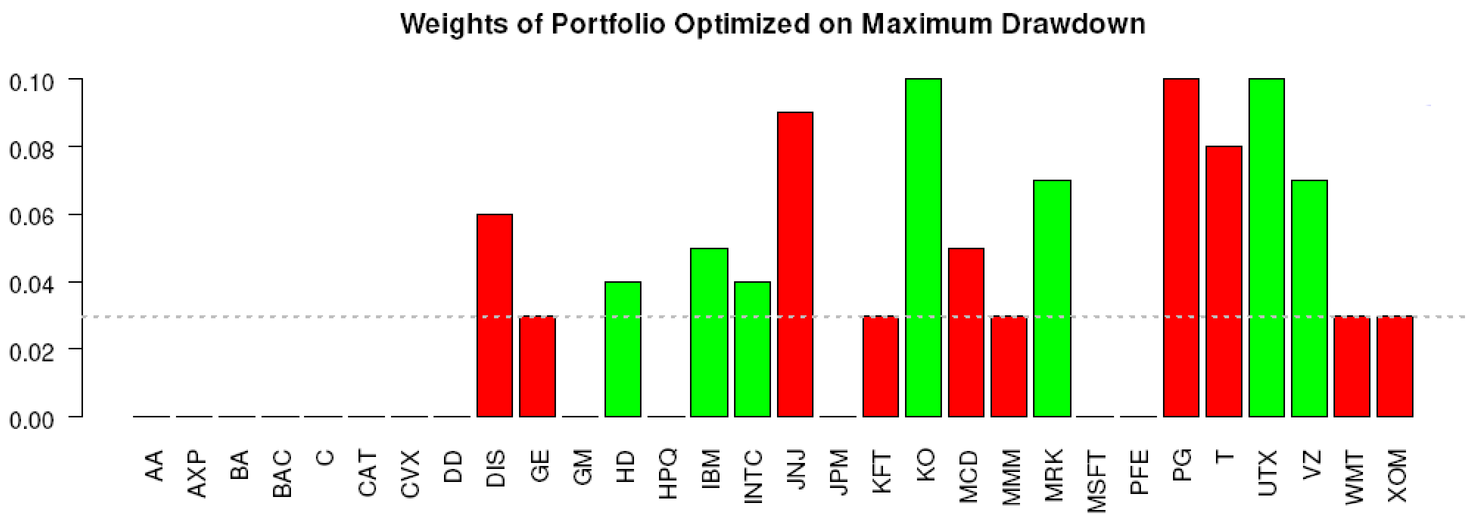

IBM: 12/02/2008 - 04/15/2009
stock price
80
85
90
95
100
Jan
Mar
Maximum
Drawdown


IBM Underwater Graph
drawdown (%)
0
-5
-10
-15
Jan
Mar

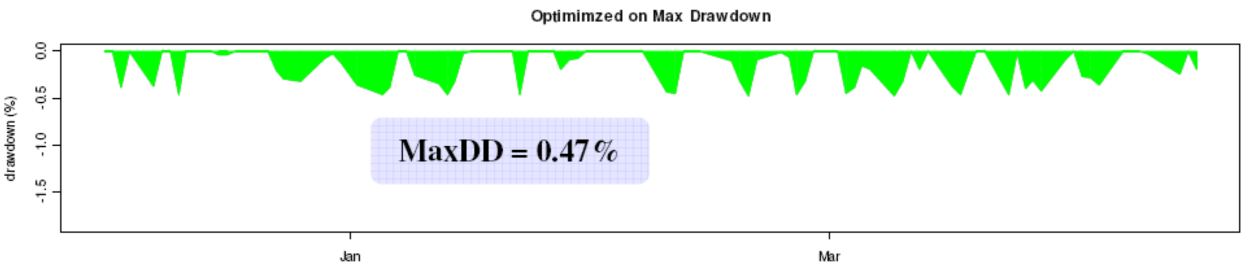
最大跌幅投资组合优化

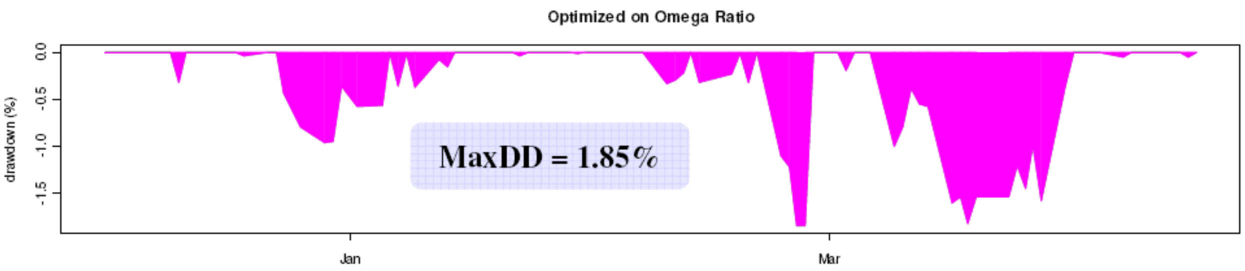
```
# max drawdown with non-zero weights between 3% & 10%
optMDD.gt3 = function(x,ret)
{
retu = ret %*% x
obj = mddx(retu,1)
weight.penalty = 100*(1-sum(x))^2
small.weight.penalty = 100*sum(x[x<0.03])
return( obj + weight.penalty + small.weight.penalty )
}
res = DEoptim(optMDD.gt3,lower,upper,
control=list(NP=2000,itermax=1000,F=0.2,CR=0.8),
ret=coredata(r))
```

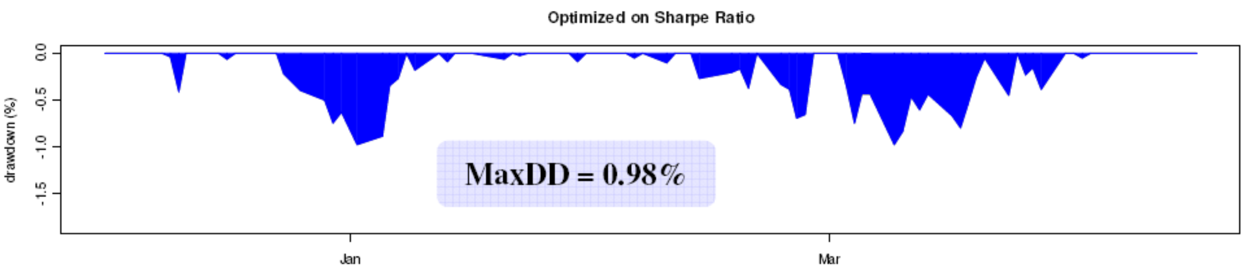
最大跌幅投资组合优化


Weights of Portfolio Optimized on Maximum Drawdown
0.10
0.08
0.06
0.04
0.02
0.00
AA
AXP
BA
BAC
C
CAT
CVX
DD
DIS
GE
GM
HD
HPQ
IBM
INTC
JNJ
JPM
KFT
KO
MCD
MMM
MRK
MSFT
PFE
PG
T
UTX
VZ
WMT
XOM

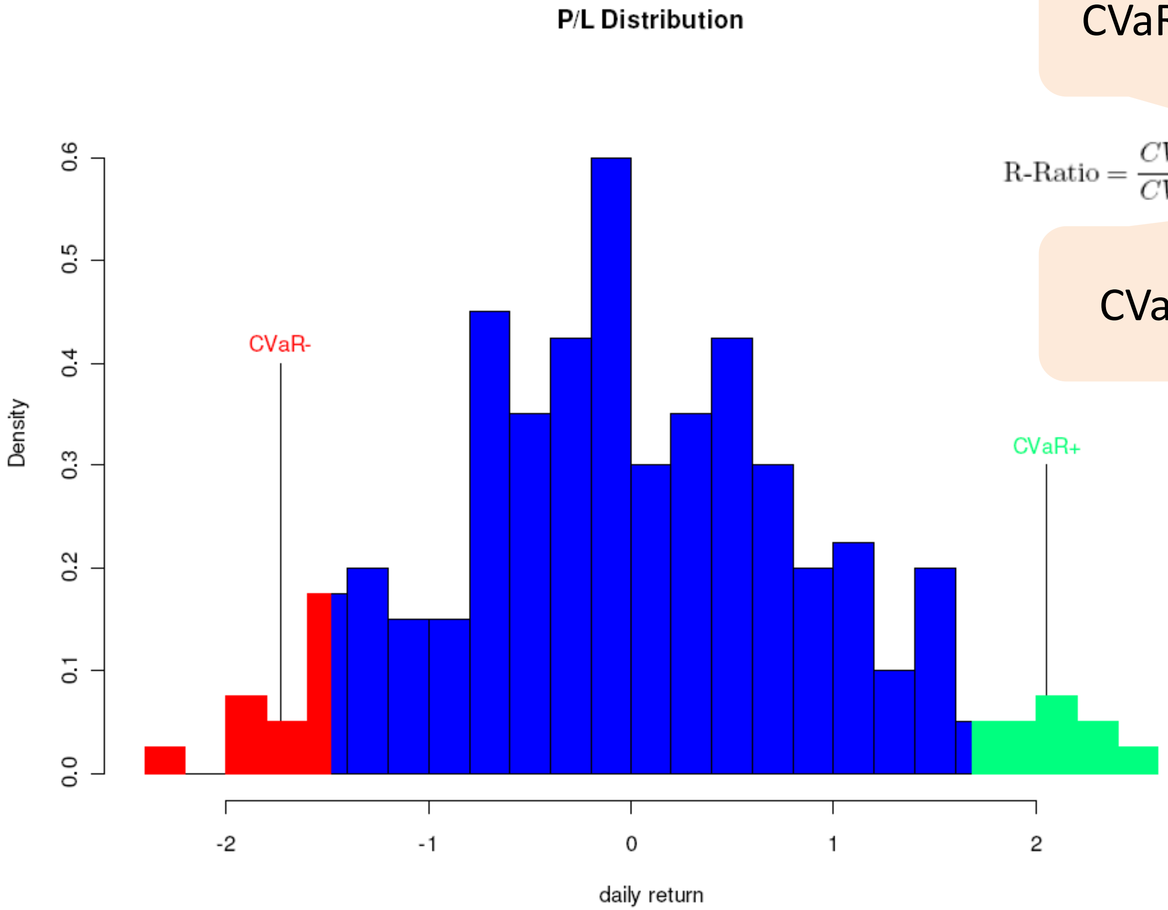
三种方法下的MaxDD对比







Rachev Ratio(R-Ratio)投资组合优化



$CVaR(\alpha,r)^{+}=-CVaR(\alpha,-r)$

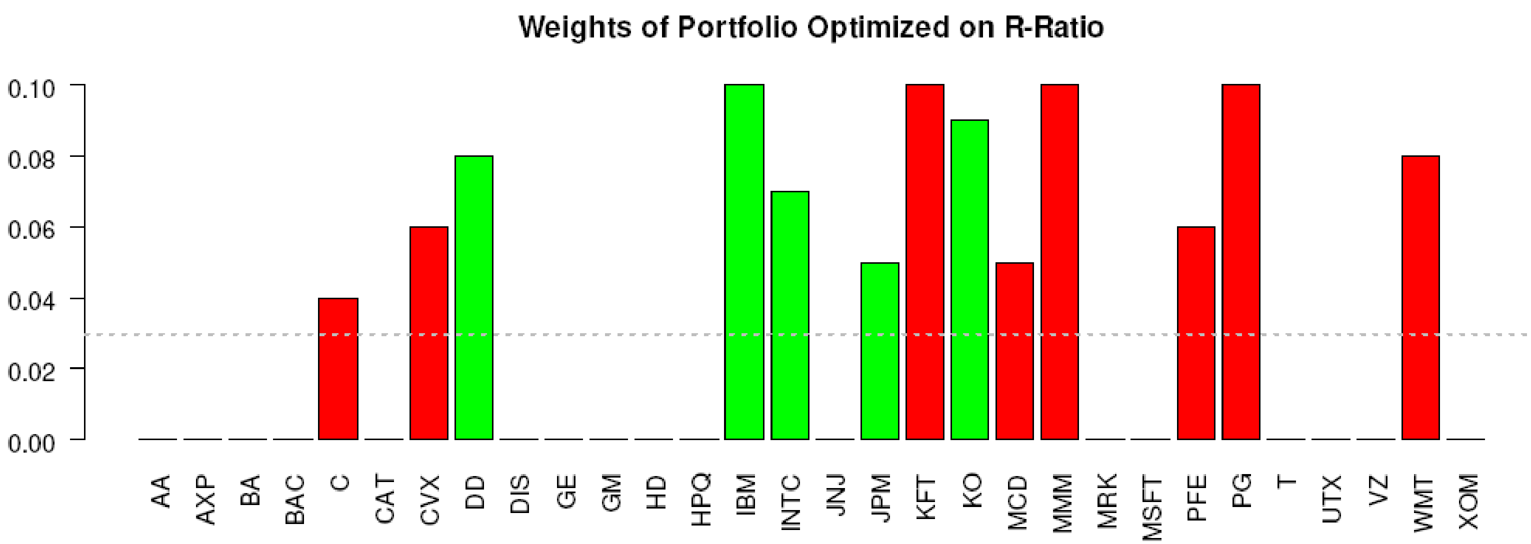
$$\text{R-Ratio} = \frac{CVaR(\alpha)^{+}}{CVaR(\alpha)^{-}}$$

$CVaR(\alpha,r)^{-}=CVaR(\alpha,r)$

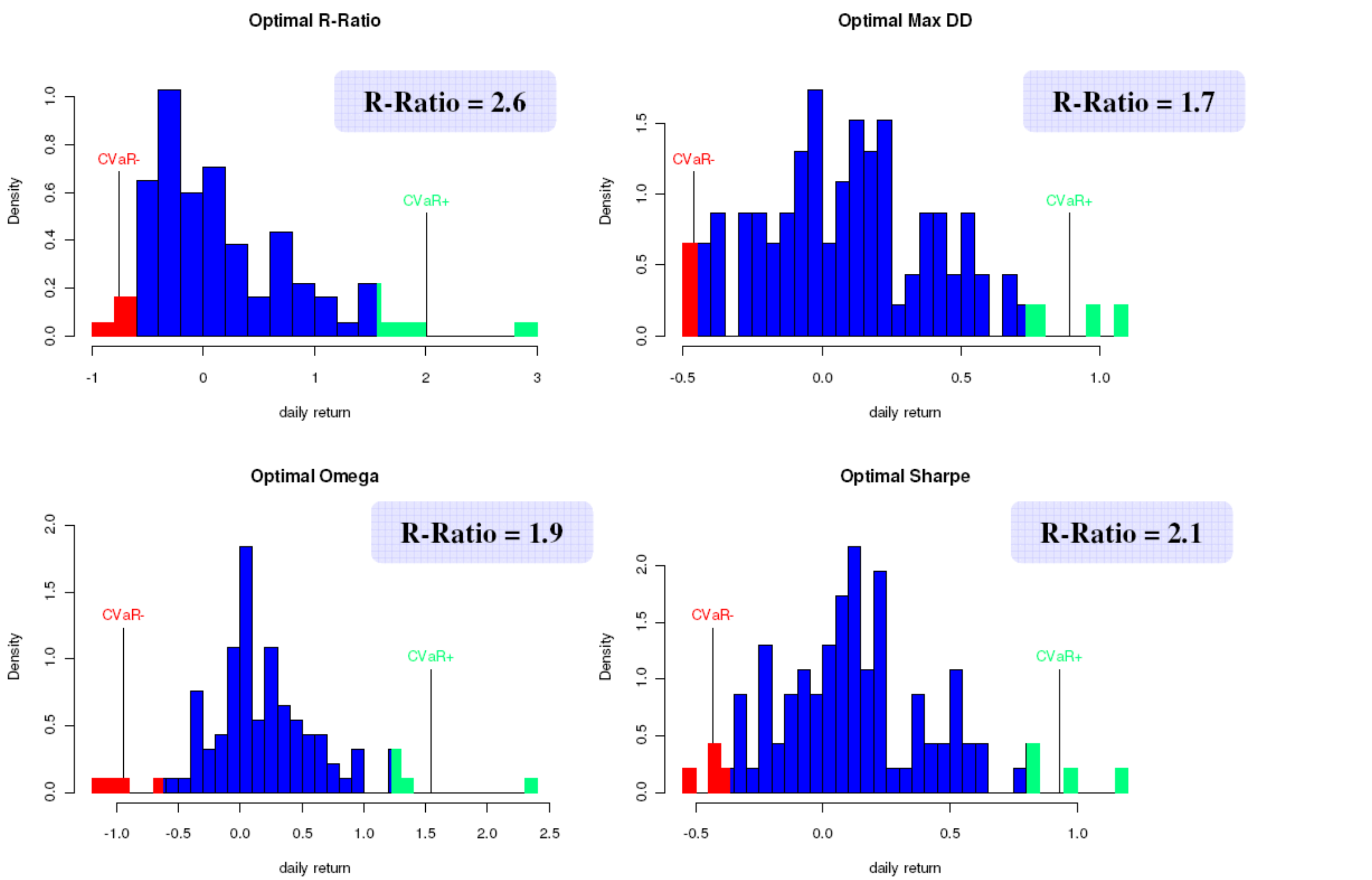
R-Ratio投资组合优化

```
optRR.gt3= function(x,ret)
{
retu = ret %*% x
obj = -CVaR(-retu)/CVaR(retu)
weight.penalty = 100*(1-sum(x))^2
small.weight.penalty = 100*sum(x[x<0.03])
return( obj + weight.penalty + small.weight.penalty )
}
res = DEoptim(optRR.gt3,lower,upper,
control=list(NP=2000,itermax=1000,F=0.2,CR=0.8),
ret=coredata(r))
```

R-Ratio投资组合优化



四种优化方法的R-Ratio对比图



相关R资料

- Portfolio Optimization with R/Rmetrics
- Differential Evolution (DEoptim) for Non-Convex Portfolio Optimization
- R Tools for Portfolio Optimization
- Large-scale portfolio optimization with DEoptim