MATLAB

```
function [Er,Sigma,Wstar,Erp,Sigmap]=Portfolioopt(P,m)
r=(P(2:end,:)-P(1:end-1,:))./P(1:end-1,:);
Er=mean(r)';
Sigma=cov(r);
N=length(Er);
e=ones(N,1);
k1=e'*(Sigma^(-1))*Er;
k2=e'*(Sigma^(-1))*e;
k3=Er'*(Sigma^(-1))*Er;
k4=Er'*(Sigma^(-1))*e;
Wstar=zeros(N,m);
Erp=zeros(m,1);
Sigmap=zeros(m,1);
Wstar(:,1)=Sigma^(-1)*e*k2^(-1);
Erp(1,1)=Wstar(:,1)'*Er;
Sigmap(1,1)=Wstar(:,1)'*Sigma*Wstar(:,1);
Wstar(:,m)=(Er==max(Er));
Erp(m,1)=Wstar(:,m)'*Er;
Sigmap(m,1)=Wstar(:,m)'*Sigma*Wstar(:,m);
selr=linspace(Erp(1,1),Erp(m,1),m);
for i=2:m-1
    Lambda1=k1^(-1)-(selr(i)*k3^(-1)-k1^(-
1))*(k1*k3^(-1)-k2*k1^(-1))^(-1)*k2*k1^(-1);
    Lambda2=(selr(i)*k3^(-1)-k1^(-1))*(k1*k3^(-1)-
k2*k1^(-1))^(-1);
```

Scanned with CamScanner

```
    Wstar(:,i)=Lambda1*(Sigma^(-
1))*Er+Lambda2*(Sigma^(-1))*e;
    Erp(i,1)=Wstar(:,i)'*Er;
    Sigmap(i,1)=Wstar(:,i)'*Sigma*Wstar(:,i);
end
```

نام فایل: Portfolioopt

توضیحات:

Er: ماتریس بازده مورد انتظار،

Sigma: ماتریس واریانس- کوواریانس،

Wstar: وزن‌های بهینه شده،

Erp: ماتریس بازده پرتفوی روی مرز کارا،

Sigmap: ماتریس واریانس پرتفوی روی مرز کارا،

P: ماتریس قیمت،

m: تعداد پرتفوی روی مرز کارا.

نکات:

- دستور Wstar(:,1)=... برای محاسبه وزن پرتفوی دارای کمترین واریانس استفاده شده است.
- دستور Erp(1,1)=... برای محاسبه بازدهی پرتفوی دارای کمترین واریانس و دستور Sigmap(1,1)=... برای محاسبه واریانس این پرتفوی استفاده شده است.
- دستور selr=... برای بخش‌بندی بازدهی پرتفوی به m قسمت مساوی است.
- در حلقه for، برای نقاط مختلف بر روی مرز کارا، عمل بهینه‌سازی تکرار می‌شود.

Scanned with CamScanner

```
clear
clc
load MonthlyIndices
k=size(Names,1);
m_r=zeros(k-1,1);
sigma_r=zeros(k-1,1);
beta_r=zeros(k-1,1);
E_r=zeros(k-1,1);
```

Scanned with CamScanner

```
r_m=price2ret(Data.TotalIndex);
m_rm=mean(r_m);
sigma_rm=std(r_m);
for i=1:k-1
    tmp=eval(['Data.' Names{i}]);
    r_tmp=price2ret(tmp);
    m_r(i,1)=mean(r_tmp);
    sigma_r(i,1)=std(r_tmp);
    c=cov(r_tmp,r_m);
    beta_r(i,1)=c(1,2)/c(2,2);
    E_r(i,1)=rf+(m_rm-rf)*beta_r(i,1);
end
plot(beta_r,E_r);
hold on
plot(beta_r,E_r,'ko');
plot(beta_r,m_r,'*r')
for j=1:k-1
    text(beta_r(j,1),m_r(j,1),['\leftarrow' Names{j}])
end
result=[m_r,E_r,beta_r,sigma_r;m_rm,m_rm,1,sigma_rm];
```

نام فایل: SML

توضیحات:

m_r: متوسط بازدهی تاریخی هر دارایی،

E_r: بازدهی مورد انتظار براساس مدل قیمت‌گذاری دارایی سرمایه‌ای هر دارایی،

beta_r: مقدار بتای هر دارایی،

sigma_r: انحراف معیار بازدهی تاریخی هر دارایی،

m_rm: متوسط بازدهی تاریخی پرتفوی بازار،

sigma_rm: انحراف معیار بازدهی تاریخی پرتفوی بازار،

rf: نرخ بازدهی بدون ریسک.

نکات:

- با اجرای دستور load MonthlyIndices داده‌های ماهانه ده شاخص صنعت بازیابی می‌شود.
- در حلقه for اول، مقدار بتا برای هر صنعت تخمین زده می‌شود. در این حلقه دستور tmp=eval... برای انتخاب داده‌های مربوط به هر صنعت، براساس نام آن صنعت است. از آنجا که ماتریس نام صنایع یک ماتریس از نوع داده‌های متنی است، برای انتخاب هر درایه از این ماتریس از علامت کروشه استفاده شده است (Names{i}). همچنین دستور eval برای اجرای یک عبارت، از جنس متن استفاده شده است.
- در حلقه دوم نام هر صنعت با استفاده از یک فلش (با دستور '\leftarrow')، در کنار نقطه مربوطه در نمودار نمایش داده می‌شود.

Scanned with CamScanner

```
function [S_r,T_r,J_A,I_r,M2]=PA(P,Index,rf)
r=price2ret(P);
rm=price2ret(Index);
m_r=mean(r);
m_rm=mean(rm);
c=cov(r,rm);
sigma_r=c(1,1)^.5;
sigma_rm=c(2,2)^.5;
Beta=c(1,2)/c(2,2);
S_r=(m_r-rf)/sigma_r;
T_r=(m_r-rf)/Beta;
J_A=m_r-(rf+(m_rm-rf)*Beta);
I_r=J_A/(std(r-(rf+(rm-rf)*Beta)));
r_P_star=sigma_rm/sigma_r*m_r+(1-sigma_rm/sigma_r)*rf;
M2=r_P_star-m_rm;
```

Scanned with CamScanner

نام فایل: PA

توضیحات:

S_r: نسبت شارپ،

T_r: نسبت ترینر،

J_A: آلفای جنسن،

I_r: نسبت اطلاعات،

M2: معیار M^2 ،

P: ارزش پرتوفوی مورد بررسی،

Index: شاخص بازار،

rf: نرخ بازدهی بدون ریسک.

Scanned with CamScanner