**Codes used:**

%Input:

%assets={'MSFT','MSFT','MSFT','MSFT','MSFT','MSFT','MSFT','MSFT','MSFT','MSFT'};

%InvAmount=[100,200,300,400,500,600,700,800,900,1000];

assets={'BMW.DE','VOW.DE','PAH3.DE','PEU.DE'};

InvAmount=[100,100,100,-100];

n=length(assets);

start\_date = '01042019';

end\_date = '01042021';

alpha=0.05;

%--Calculate returns--------

Dates=(datenum(str2num(start\_date(5:8)), ...

str2num(start\_date(3:4)),str2num(start\_date(1:2))):...

datenum(str2num(end\_date(5:8)),str2num(end\_date(3:4)),...

str2num(end\_date(1:2))))';

for i=1:n

stocks = hist\_stock\_data(start\_date, end\_date, char(assets(i)));

% matlab date conversion

stocks.Date;

stocks.Date=datenum(stocks.Date);

Until here it is created for the figure 1

% reverse vectors old -> new

stocks.Date = stocks.Date(end:-1:1);

stocks.Close = stocks.Close(end:-1:1);

stocks.Open = stocks.Open(end:-1:1);

stocks.High = stocks.High(end:-1:1);

stocks.Low = stocks.Low(end:-1:1);

stocks.Volume = stocks.Volume(end:-1:1);

stocks.AdjClose = stocks.AdjClose(end:-1:1);

%Delete irrelevant dates and merge data

if i==1

Dates=Dates(ismember(Dates,stocks.Date,'rows'),:);

Prices=horzcat(Dates,stocks.Close);

else

stocks.Close=stocks.Close(ismember(stocks.Date,Prices(:,1),'rows'),:);

Prices=Prices(ismember(Prices(:,1),stocks.Date,'rows'),:);

Prices=horzcat(Prices,stocks.Close);

end

end

Until here it is created for the figure 2

Prices;

ret = Prices(2:end,2:n+1)./Prices(1:end-1,2:n+1)-1;

'length'

length(ret)

corr(ret)

%----Create all possible coalitions

S=coalitions(n);

VAR=zeros(2^n-1,1);

ES=zeros(2^n-1,1);

for i=1:2^n-1

S1=S(i,:);

XS=ret\*(S1.\*InvAmount)';

VAR(i)=-quantile(XS,alpha);

ES(i)=-mean(XS(XS<=quantile(XS,alpha)));

End

Until here it is created for the figure 3

Activity\_ES=activity(S,ES);

Incremental\_ES=incremental(S,ES);

Cost\_gap\_ES=cost\_gap(S,ES);

Beta\_ES=beta(ret,InvAmount,ES);

Shapley\_ES=shapley(S,ES);

Nucleolus\_ES=nucleolus(S,ES);

Lorenz\_ES=lorenz(S,ES);

Activity\_VAR=activity(S,VAR);

Incremental\_VAR=incremental(S,VAR);

Cost\_gap\_VAR=cost\_gap(S,VAR);

Beta\_VAR=beta(ret,InvAmount,VAR);

Shapley\_VAR=shapley(S,VAR);

Nucleolus\_VAR=nucleolus(S,VAR);

Lorenz\_VAR=lorenz(S,VAR);

[AS\_VAR,AS\_ES]=as(S,ret,InvAmount, alpha);

Solution\_ES=[Activity\_ES';Incremental\_ES';Beta\_ES';Cost\_gap\_ES';Nucleolus\_ES';Shapl

ey\_ES';Lorenz\_ES'; AS\_ES']'

sum\_ES= sum(Solution\_ES)

Solution\_VAR=[Activity\_VAR';Incremental\_VAR';Beta\_VAR';Cost\_gap\_VAR';Nucleolus\_VAR'

;Shapley\_VAR';Lorenz\_VAR';AS\_VAR']'

sum\_VaR=sum(Solution\_VAR)

Until here it is created for the figure 4

**Allocation rule**

function Activity=activity(S,rm)

n=length(S(1,:));

Activity=rm(1:n)/sum(rm(1:n)).\*rm(end);

function Incremental=incremental(S,rm)

n=length(S(1,:));

for i=1:n

Incremental(i)=rm(end)-rm(end-i);

end

Incremental=(Incremental/sum(Incremental).\*rm(end))';

function Beta=beta(ret,InvAmount,rm)

n=length(ret(1,:));

COV=zeros(n,1);

for i=1:n

COV1=cov(InvAmount(i)\*ret(:,i),ret\*InvAmount');

COV(i)=COV1(1,2);

End

Until here it is created for the figure 5

COV;

B=COV/var(ret\*InvAmount');

Beta=B/sum(B)\*rm(end);

function Cost\_gap=cost\_gap(S,rm)

n=length(S(1,:));

temp1=zeros(2^n-1,1);

gamma1=zeros(n,1);

Cost\_gap=zeros(n,1);

for i=1:length(S(:,1)) %coalitions

temp=0;

for j=1:n %players

if S(i,j)==1

temp=temp+rm(end)-rm(end-j);

end

end

temp1(i)=temp;

end

temp2=rm-temp1;

for i=1:n

index=find(S(:,i)>0);

gamma1(i)=min(temp2(index));

end

for i=1:n

Cost\_gap(i)=rm(end)-rm(end-i);

end

if sum(gamma1)~=0

for i=1:n

Cost\_gap(i)=Cost\_gap(i)+gamma1(i)/sum(gamma1)\*(rm(end)-temp1(end));

end

end

Until here it is created for the figure 6,7

function Nucleolus=nucleolus(S,rm)

prec=0.00001;

n=length(S(1,:));

temp1=zeros(2^n-1,1);

temp2=zeros(1,n+2);

temp2(end)=1;

S\_eq=ones(1,n+2);

S\_eq(end-1:end)=0;

S\_eq=vertcat(S\_eq,temp2);

S\_ineq=horzcat(S,temp1);

S\_ineq=S\_ineq(1:end-1,:);

S\_ineq2=S\_ineq;

S\_ineq2(:,end)=1;

S\_ineq=vertcat(S\_ineq,S\_ineq2);

S\_ineq=horzcat(S\_ineq,vertcat(zeros(2^n-2,1),-rm(1:end-1)));

b\_eq=vertcat(rm(end),[1]);

f=zeros(1,n+2);

f(end-1)=-1;

b\_ineq=vertcat(rm(1:end-1),zeros(2^n-2,1));

options=optimset('display','off','MaxIter',1000);

% f

% S\_ineq

% b\_ineq

% S\_eq

% b\_eq

[x,fval,exitflag]=linprog(f,S\_ineq,b\_ineq,S\_eq,b\_eq,[],[],[],options);

% x

stop1=0;

while stop1==0

q=rm-S\*x(1:end-2);

x\_copy=x;

index1=find(abs(q+fval)<prec);

add\_eq=S(index1,:);

add\_eq(:,end+1)=1;

add\_eq=horzcat(add\_eq,-rm(index1));

S\_eq=vertcat(S\_eq,add\_eq);

add\_b\_eq=zeros(length(index1),1);

b\_eq=vertcat(b\_eq,add\_b\_eq);

[x,fval,exitflag]=linprog(f,S\_ineq,b\_ineq,S\_eq,b\_eq,[],[],[],options);

if abs(x-x\_copy)<prec

stop1=1;

end

end

Until here it is created for the figure 8,9,10