```
%Yi Li June 2022
% Import Data
Data_path = 'C:\...\Copula\Data\';
Data = xlsread([Data_path,'changes.xlsx'],'Changes', 'A2:E648');
Dates = Data(:,1);
Returns = Data(:,2:end);
T = length(Returns);  % 647
N = size(Returns, 2);  % 4

% Statistic Table
Statist = nan(4,N);
Statist(1,:) = mean(Returns);
Statist(2,:) = std(Returns);
Statist(3,:) = skewness(Returns);
Statist(4,:) = kurtosis(Returns);

% Correlation Table
Corr = zeros(N,N);
Corr = corrcoef(Returns);

%Visualization
figure(1),subplot(2,2,1),plot((1:T)',Data(:,2),'b-','LineWidth',1);
title(' Series 1')
hold on;
subplot(2,2,2),plot((1:T)',Data(:,3),'b-','LineWidth',1);
title('Series 2')
hold on;
subplot(2,2,3),plot((1:T)',Data(:,4),'b-','LineWidth',1);
title('Series 3')
hold on;
subplot(2,2,4),plot((1:T)',Data(:,5),'b-','LineWidth',1);
title('Series 4')
grid on;

% JB and ARCH Tests
JB = zeros(1, N);
for i = 1: N
    [h, pValue] = lbqtest(Returns(:,i),'lags',[10]);
    JB(:,i) = pValue;
end

ARCH = zeros(1, N);
for i = 1: N
    [h,pValue,stat,cValue] = archtest(Returns(:,i), lag=5);
    ARCH(:,i) = pValue;
end

% ARMA Model
residuals = nan(T, N);
mean_order = nan(N, 2);

for i=1:N
    [theta,sig,vcv,order,resids1] = ARMAX_opt(Returns(:, i),5,5,'AIC');  % takes↙
about 6 seconds per variable
```

```
    mean_order(i, 1:2) = order';
    residuals(:, i) = [zeros(max(order), 1);resids1]; % fill in max order1 zeros
end

% EGARCH Model
Vol = nan(T, N);
EGARCH = egarch('GARCHLags',1:2,'ARCHLags',1:2);
Mdl.Distribution = "Gaussian";
for i= 1: N
     ModelEst = estimate(EGARCH, residuals(:, i));
     voli = infer(ModelEst, residuals(:, i));
     Vol(:, i) = voli;
end

% Retest Standardized Series
stdresids = residuals./sqrt(Vol);

Retest_JB = zeros(1, N);
for i = 1: N
    [h, pValue] = lbqtest(stdresids(:,i),'lags',[10]);
    Retest_JB(:,i) = pValue;
end

RE_ARCH = zeros(1, N);
for i = 1: N
    [h,pValue,stat,cValue] = archtest(stdresids(:,i), lag=5);
    RE_ARCH(:,i) = pValue;
end

% SKEWT Distribution
options = optimset('Display','off','TolCon',10^-12,'TolFun',10^-4,'TolX',10^-↙
6,'DiffMaxChange',Inf,'DiffMinChange',0,'Algorithm','active-set');
outSKEWT = nan(N,2); %dof and asymmetry parameters
lower = [2.1, -0.99];
upper = [Inf, 0.99 ];
theta0 = [6;0];
for i = 1: N
    theta1 = fmincon('skewtdis_LL',theta0,[],[],[],[],lower,upper,[],options,↙
stdresids(:,i));
    outSKEWT(i,:) = theta1';
end

Uskewt = nan(T,N);
for i=1: N
    Uskewt(:, i) = skewtdis_cdf(stdresids(:,i),outSKEWT(i,1),outSKEWT(i,2));
end

% Visualization
figure(1),subplot(2,2,1),hist(Uskewt(:,1));
title('Series 1')
hold on;
subplot(2,2,2),hist(Uskewt(:,2));
title('Series 2')
hold on;
```

```
subplot(2,2,3),hist(Uskewt(:,3));
title('Series 3')
hold on;
subplot(2,2,4),hist(Uskewt(:,4));
title('Series 4')
grid on;

% Test Uniform
pd = makedist('Uniform',0,1);
KS = nan(1, N);
for i = 1:N
    Utest = kstest(Uskewt(:,i),'CDF',pd);
    KS(:, i) = Utest;
end

% Copula Models
tau_tail = nan(5,2);
opt_copula_model= nan(1,N-1);
i=1;
for j=2:N
    u=Uskewt(:,i);
    v=Uskewt(:,j);

    options = optimset('Display','iter','TolCon',10^-12,'TolFun',10^-4,'TolX',10^-↙
6);
    % 1. Normal Copula
    kappa1 = corrcoef12(norminv(u),norminv(v));
    LL1 = NormalCopula_CL(kappa1,[u,v]);

    % 2. Rotated Clayton copula
    lower = 0.0001;
    theta0 = 1;
    [ kappa2 LL2] = fmincon('claytonCL',theta0,[],[],[],[],lower,[],[],options,1-[u,↙
v]);

    % 3. Gumbel copula
    lower = 1.1;
    theta0 = 2;
    [ kappa3 LL3] = fmincon('gumbelCL',theta0,[],[],[],[],lower,[],[],options,[u,↙
v]);

    % 4. Student's t copula
    lower = [-0.9 , 2.1 ];
    upper = [ 0.9 , 100 ];
    theta0 = [kappa1;10];
    [ kappa4 LL4] = fmincon('tcopulaCL',theta0,[],[],[],[],lower,upper,[],options,↙
[u,v]);

    % 5. Symmetrised Joe-Clayton copula
    lower = [0 , 0 ];
    upper = [ 1 , 1];
    theta0 = [0.25;0.25];
    [ kappa5 LL5] = fmincon('sym_jc_CL',theta0,[],[],[],[],lower,upper,[],options,↙
[u,v]);
```

```
    LL = [LL1;LL2;LL3;LL4;LL5];
    opt = find(LL==min(LL));
    opt_copula_model(:,j-1) = opt;

    %Tail dependence
    tau_tail(1,:,j-1) = [0,0];                  % Normal copula: zero upper and lower
    tau_tail(2,:,j-1) = [0,2^(-1/kappa2)];      % Rotated Clayton copula: zero lower
    tau_tail(3,:,j-1) = [0,2-2^(1/kappa3)];     % Gumbel copula: zero lower
    tau_tail(4,:,j-1) = ones(1,2)*2*tdis_cdf(-sqrt((kappa4(2)+1)*(1-kappa4(1))/
(1+kappa4(1))),kappa4(2)+1);  % Student's t copula: symmetric dependence
    tau_tail(5,:,j-1) = kappa5([2,1])';               % SJC copula parameters: upper
and lower
end
```