```
function [diagnostics]=klausdiagnostics(M);

%this version adds  pr(>0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Accuracy assessment and convergence diagnostics for GS output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% input:  a k by R matrix of GS draws, where k is the number of parameters
% and r is the number of draws (after discarding burn-ins)

% output: "diag" with
%       column 1 = means
%       column 2 = std's
%       column 3 = nse (autocorrelation-adjusted)
%       column 4 = IEF (inefficiency factors - see meta_BMA / JAE notes
%                       and Koop, Poirier & Tobias p. 144
%       column 5 = effective sample size (i.e. how many draws would achieve
%                  the same accuracy under iid conditions)
%       column 6 = Geweke's (1992) CD diagnostics based on AC-adjusted
%                  nse's

k=size(M,1);
R=size(M,2);

% Autocorrelation, nse, and efficiency
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

IEF=zeros(k,1);
nse=zeros(k,1);

i=1;
for i=1:k
   int=M(i,:)'; % pick draws for a single parameter
   intsum=0;
   j=1;
   for j=1:(R-2)
       int1=int(1:R-j,1);
       int2=int(j+1:R,1);
       int3=corrcoef(int1,int2);
%        if isnan(int3(1,2));
%            int3(1,2)=0;
%        end;
       intsum=intsum+(1-j/R)*int3(1,2);
       if abs(int3(1,2))<0.05
           break % exit innermost loop
       end
   end
   intsum=max(0,intsum); %for situations when the loop immediately cuts off
   % with a slight negative value for the correlation
   IEF(i,1)=1+2*intsum;
end
```

```
nse=sqrt(((1/R)*(var(M')'.*IEF)));
mstar=R./IEF;

% CD diagnostics

g1=round(0.1*R);
g2=round(0.6*R)+1;
M1=M(:,1:g1);
M2=M(:,g2:end);
R1=size(M1,2);
R2=size(M2,2);

mM1=mean(M1,2);
mM2=mean(M2,2);

IEF1=zeros(k,1);
nse1=zeros(k,1);
i=1;
for i=1:k
   int=M1(i,:)'; % pick draws for a single parameter
   intsum=0;
   j=1;
   for j=1:(R1-2)
       int1=int(1:R1-j,1);
       int2=int(j+1:R1,1);
       int3=corrcoef(int1,int2);
%        if isnan(int3(1,2));
%            int3(1,2)=0;
%        end;
       intsum=intsum+(1-j/R1)*int3(1,2);
       if abs(int3(1,2))<0.05
           break % exit innermost loop
       end
   end
   intsum=max(0,intsum); %for situations when the loop immediately cuts off
   % with a slight negative value for the correlation
   IEF1(i,1)=1+2*intsum;
end
nse1=sqrt(((1/R1)*(var(M1')'.*IEF1)));

IEF2=zeros(k,1);
nse2=zeros(k,1);
i=1;
for i=1:k
   int=M2(i,:)'; % pick draws for a single parameter
   intsum=0;
   j=1;
   for j=1:(R2-2)
       int1=int(1:R2-j,1);
       int2=int(j+1:R2,1);
       int3=corrcoef(int1,int2);
%        if isnan(int3(1,2));
%            int3(1,2)=0;
%        end;
```

```
       intsum=intsum+(1-j/R2)*int3(1,2);
       if abs(int3(1,2))<0.05
           break % exit innermost loop
       end
   end
   intsum=max(0,intsum); %for situations when the loop immediately cuts off
   % with a slight negative value for the correlation
   IEF2(i,1)=1+2*intsum;
end
nse2=sqrt(((1/R2)*(var(M2')'.*IEF2)));

CD=(mM1-mM2)./(sqrt(nse1.^2+nse2.^2));

%Pr(>0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%prob>0
p0vec=zeros(k,1);
for i=1:k
    p0vec(i)=length(find(M(i,:)>0))/R;
end

diagnostics=[mean(M,2) std(M')' p0vec nse IEF mstar CD];
```

Published with MATLAB® R2022b