Code to convert BNs variables into Level 3 FCs weights

```
function FinalHeirarchy_Averaged()
%%
Level2Data = readtable('Combined data UC-I_AHP GM_with
Intersectional', 'Sheet', 'Hierarchy');
[file, path] = uigetfile;
selectedfile = fullfile(path,file); % Select Hierarchy file from BN
results.
Level3Weights = readtable(selectedfile);
%%
L2D = table2dataset(Level2Data);
L3W = table2array(Level3Weights);


%% Writing the BN weight obtained by Julia into the final result file.
% IF loop compared the code of variables from selected Julia .csv and
% Matlab excel and write the weight of right variable to right place
for i = 1:width(L3W)
    for l = 1:height(L2D)
        if strcmp(L2D{l,8}, Level3Weights.Properties.VariableNames{i})
            L2D{l,9} = L3W(4,i);
        end
    end

end

%% Writing above weight back to the result file.
xlswrite('Combined data UC-I_AHP GM_with Intersectional.xlsm',
dataset2cell(L2D(:,9)), 'Hierarchy', 'I1');

%% Loading the weight and variable data again from hierarchy sheet.
Level3Data = readtable('Combined data UC-I_AHP GM_with
Intersectional', 'Sheet', 'Hierarchy', 'TextType', 'string');
%%
L3D1 = table2array(Level3Data(:,5));
L3D2 = table2array(Level3Data(:,6));
L3D3 = table2array(Level3Data(:,9));
L3D4 = table2array(Level3Data(:,10));
clear isnan
L3D3(isnan(L3D3)) = 0;
%%
A = [L3D2 L3D3 L3D4];
A = rmmissing(A);
%%
VarName = char(A(:,1));
FinalW = str2double(A(:,2));
NormW = str2double(A(:,3));
FCName = L3D1;
%% Separating uniqe and duplicate variable.
%% Separating uniqe and duplicate variable.
% Separating variables with five characters
k = 1;
for i = 1:length(VarName)
    if VarName(i, 7) == ' ' && VarName(i, 6) == ' '
        UniqVar5chr(k, :) = VarName(i, :);
        UniqFinalW5chr(k, :) = FinalW(i, :);
        UniqNormW5chr(k, :) = NormW(i, :);
        UniqFCName5chr(k, :) = FCName(i, :);
        k = k+1;
```

```
    end
end
%% Reassembling variable code and their wieght
UniqMat5chr = [UniqFCName5chr string(UniqVar5chr) UniqFinalW5chr
UniqNormW5chr];

%% Separating variables with seven characters
clear i k
k = 1;
for i = 1:length(VarName)
    if VarName(i, 7) == 'a' || VarName(i, 7) == 'b' || VarName(i, 7)
== 'c' || VarName(i, 7) == 'd' || VarName(i, 7) == 'e' || VarName(i,
7) == 'f' || VarName(i, 7) == 'g'
        MultiVar7chr(k, :) = VarName(i, :);
        MultiFinalW7chr(k, :) = FinalW(i, :);
        MultiNormW7chr(k, :) = NormW(i, :);
        MultiFCName7chr(k, :) = FCName(i, :);
        k = k+1;
    end
end
%% Separating duplicate variables with six charaters
clear i k
k = 1;
for i = 1:length(VarName)
    if VarName(i, 6) == 'a' || VarName(i, 6) == 'b' || VarName(i, 6)
== 'c' || VarName(i, 6) == 'd' || VarName(i, 6) == 'e' || VarName(i,
6) == 'f' || VarName(i, 6) == 'g' || VarName(i, 6) == 'h' ||
VarName(i, 6) == 'i' || VarName(i, 6) == 'j' || VarName(i, 6) == 'k'
|| VarName(i, 6) == 'l' || VarName(i, 6) == 'm' || VarName(i, 6) ==
'n'
        MultiVar6chr(k, :) = VarName(i, :);
        MultiFinalW6chr(k, :) = FinalW(i, :);
        MultiNormW6chr(k, :) = NormW(i, :);
        MultiFCName6chr(k, :) = FCName(i, :);
        k = k+1;
    end
end
%% Separating uniqe variables with six characters.
clear i k
k = 1;
for i = 1:length(VarName)
    if VarName(i, 7) == ' ' && VarName(i, 6) ~= ' ' && VarName(i, 6)
~= 'a' && VarName(i, 6) ~= 'b' && VarName(i, 6) ~= 'c' && VarName(i,
6) ~= 'd' && VarName(i, 6) ~= 'e' && VarName(i, 6) ~= 'f' &&
VarName(i, 6) ~= 'g' && VarName(i, 6) ~= 'h' && VarName(i, 6) ~= 'i'
&& VarName(i, 6) ~= 'j' && VarName(i, 6) ~= 'k' && VarName(i, 6) ~=
'l' && VarName(i, 6) ~= 'm' && VarName(i, 6) ~= 'n'
        UniqVar6chr(k, :) = VarName(i, :);
        UniqFinalW6chr(k, :) = FinalW(i, :);
        UniqNormW6chr(k, :) = NormW(i, :);
        UniqFCName6chr(k, :) = FCName(i, :);
        k = k+1;
    end
end
%% Reassembling variable code and their wieght
UniqMat6chr = [UniqFCName6chr string(UniqVar6chr) UniqFinalW6chr
UniqNormW6chr];

%% Calculation of Avarage weight of seven character duplicate
[Cm7,iam7,idxm7] = unique(str2num(MultiVar7chr(:, (3:6))),'stable');
valFinalW7chr = accumarray(idxm7,MultiFinalW7chr(:,1),[],@mean);
```

```
valNormW7chr = accumarray(idxm7,MultiNormW7chr(:,1),[],@mean);
S = string({'FC'});
VarCm7chr =  S + Cm7 (:);
MultiFCName7chr = rmmissing(MultiFCName7chr);
AvMat7chr = [MultiFCName7chr VarCm7chr valFinalW7chr valNormW7chr];
%% Calculation of Avarage weight of six character duplicate
[Cm6,iam6,idxm6] = unique(str2num(MultiVar6chr(:, (3:5))),'stable');
valFinalW6chr = accumarray(idxm6,MultiFinalW6chr(:,1),[],@mean);
valNormW6chr = accumarray(idxm6,MultiNormW6chr(:,1),[],@mean);
%S = string({'FC'});
VarCm6chr =  S + Cm6 (:);
MultiFCName6chr = rmmissing(MultiFCName6chr);
AvMat6chr = [MultiFCName6chr VarCm6chr valFinalW6chr valNormW6chr];

%% DONE!!!!!!!!!!!!!!!!!!
Avrg = [UniqMat5chr; UniqMat6chr; AvMat6chr; AvMat7chr];
%% Making the hierarchy
[~, idx] = sort(str2double(Avrg(:, 4)), 'descend');
% Apply the sorting to the initial matrix
FinalHierarchy  = Avrg(idx, :);
%FinalHierarchy = sortrows(Avrg, 4, 'descend');
%% writing the results back to the file.
xlswrite('Combined data UC-I_AHP GM_with Intersectional.xlsm',
FinalHierarchy, 'Hierarchy', 'T2');

%% writing Heirarchy of Variables.
FCL3 = table2array(Level3Data(:,7)); % Final weight
B = [FCL3 L3D2 L3D4];
B = rmmissing(B);
% Sort the column after having casted it to double
clear idx
[~, idx] = sort(str2double(B(:, 3)), 'descend');
% Apply the sorting to the initial matrix
FinalHierarchy_Var  = B(idx, :);
%%
xlswrite('Combined data UC-I_AHP GM_with Intersectional.xlsm',
FinalHierarchy_Var, 'Hierarchy', 'N2');

end
```