

```

% The function simdec builds a decomposed probability distribution based on
% already existing Monte Carlo simulation model.

% REQUIRED INPUTS:
% 'result' is a vertical vector of a resulting variable from the Monte
% Carlo simulation,
% 'scenario' is of the same size vector that assigns scenario indeces
% (starting from 1) to corresponding 'result' values.
% ADDITIONAL INPUTS (specify empty matrix [] if not using):
% 'color' is a matrix % of the size # of scenarios by 3, that specifies
% the color displayed for each scenario (in matlab values
% http://www.cimat.mx/~max/InformaticaAplicadaII/bibliografia/MATLAB\_files/colormap.png),
% 'legend' is a horizontal vector of scenario labels to be displayed
% in the legend,
% 'axistitle' is a title for the x axis.

function b = simdec(result,scenario,color,legends,axistitle)

% separating result by scenario
result_dec=cell(1,max(scenario));
for i=1:max(scenario)
    s=size(result(scenario==i),1);
    result_dec(i)=mat2cell(result(scenario==i),s,1);
end

% defining edges of bins
l=min(result);
h=max(result);
bins=100;
edges=[l:(h-l)/bins:h];

% define frequency of each scenario NPV for each bin
f=zeros(max(scenario),100);
for i=1:max(scenario)
    f(i,:)=histcounts(cell2mat(result_dec(i)),edges);
end

% building stacked histogram
figure
b=bar(1:bins,permute(f,[2 1]),1,'stack');

% legend
if isempty(legends)==0
    legend(legends);
end
hold on

% axis titles
xlabel(axistitle);
ylabel('Probability');

% colours
if isempty(color)==0
    for i=1:max(scenario)
        b(i).FaceColor = color(i,:);
    end
end
for i=1:max(scenario)
    b(i).EdgeColor = [0.5 0.5 0.5];
end
hold on

```

```

% ticks
% real axis ticks
figure
h=histogram(result,bins);
xticks=get(gca,'xtick');
yticks=get(gca,'ytick');
close
% X axis
% getting corresponding min and max on the artificial axis
min_art=(xticks(1)-min(result))*bins/(max(result)-min(result));
max_art=(xticks(end)-max(result))*bins/(max(result)-min(result))+bins;
distance_art=max_art-min_art;
% corresponding ticks on the artificial axis
xticks_art=(xticks-xticks(1)).*distance_art./(xticks(end)-xticks(1))+min_art;
% Y axis
total=size(result,1);
yticks_art=yticks/total;
a=[cellstr(num2str(yticks_art'*100))]; % converting values into percentage
pct = char(ones(size(a,1),1)*'%'); % creating vector of % signs
new_yticks = [char(a),pct]; % add the '%' signs after the percentage values

% getting new labels to the graph
set(gca,'XLim',[min(xticks_art), max(xticks_art)],....
'XTick',xticks_art,'XTickLabel',xticks,....
'YLim',[0 yticks(end)],'YTickLabel',new_yticks);

hold off

end

```