

```
function handles = repairCollapseDemoProb(handles)

% Add all of the losses for the different story together
handles.Loss_IM = sum(handles.Loss_IM_Story);

% Calculate the probability of demo and repair
P_NC = 1 - handles.P_collapse;
handles.P_NC_demo = P_NC .* handles.demo.p_im;
handles.P_NC_repair = P_NC.*(1-handles.demo.p_im);

% Calculate the demoLoss and collapseLoss repair loss weighted
handles.demoLoss_IM = handles.P_NC_demo * handles.demolishionCost;
handles.collapseLoss_IM = handles.P_collapse * handles.collapseCost;
handles.ncLoss_IM = handles.P_NC_repair .* handles.Loss_IM;

% Calculate Loss given IM total
handles.L_IM = handles.ncLoss_IM + handles.collapseLoss_IM + handles.demoLoss_IM;
Sa = handles.hazardCurve(1,:);

% Plot the Losses given IM
figure
hold on
plot([0,Sa], [0, handles.L_IM], 'k', 'LineWidth', 1.5)
plot(Sa, handles.ncLoss_IM, '--r')
plot(Sa, handles.collapseLoss_IM, '--g')
plot(Sa, handles.demoLoss_IM, '--b')
xlabel('Sa (g)')
ylabel('E[L|IM] ($)')
title('Expected Loss Given IM')
legend('Total Loss', 'No Collapse, Repair', 'Collapse', 'Demo')
set(gca, ...
    'Box'          , 'off'          , ...
    'TickDir'      , 'out'          , ...
    'TickLength'   , [.02 .02]     , ...
    'XMinorTick'   , 'on'           , ...
    'YMinorTick'   , 'on');
grid on
hold off
```