```
% Tony Hyun Kim
% CS 224w, Problem 3.1

%============================================================
% Python code to compute the word frequencies:
%------------------------------------------------------------
% # Tony Hyun Kim
% # CS 224w, Problem 3(a):
% # Perform word count on given corpus
%
% from collections import defaultdict
%
% source = "donquijote"
% fin = open(source+".txt",'r')
%
% wordcount = defaultdict(int)
% for line in fin:
%     word = line.strip()
%     wordcount[word] += 1
%     #print word
%
% fout = open(source+"_count.txt",'w')
% for word, count in wordcount.items():
%     fout.write(str(count)+'\n')
%     #print word + " - " + str(count)
%============================================================

clear all; close all;

source = 'mobydick_count.txt';
% source = 'donquijote_count.txt';
data = load(source); % Frequency of each word
N = length(data);

% Plot the empirical distribution
%------------------------------------------------------------
x = 1:1e4;
h = hist(data,x);
p = h/N;
subplot(121);
loglog(x,p,'.');
title(strrep(source,'_','\_'));
xlabel('Word frequency');
ylabel('Fraction');
hold on;

% MLE fit
%------------------------------------------------------------
xmin  = 1; % Since we observe words with this frequency

% Formulas from the continuous distribution
alpha = 1 + N/sum(log(data/xmin));
p_cont = (alpha-1)/xmin*(x/xmin).^-alpha;

% Perform numerical MLE fit based on discrete distribution
disc_alphas = linspace(0.75*alpha, 1.25*alpha);
```

```
betas = zeros(size(disc_alphas)); % Normalization factors for the discrete distr
LLs   = zeros(size(disc_alphas)); % Log likelihood
for i = 1:length(disc_alphas)
    disc_alpha = disc_alphas(i);
    betas(i) = 1/sum((x/xmin).^-disc_alpha);
    LLs(i)   = sum(log(betas(i).*(data/xmin).^-disc_alpha));
end

subplot(122);
plot(disc_alphas,LLs);
xlim([disc_alphas(1) disc_alphas(end)]);
xlabel('\alpha');
ylabel('Discrete log likelihood');

hold on;
% Show the MLE estimate assuming continuous distribution
plot(alpha,interp1(disc_alphas,LLs,alpha,'linear'),'ro');

% Show the MLE estimate assuming discrete distribution
[~, ind] = max(LLs);
disc_alpha = disc_alphas(ind);
beta       = betas(ind);
mll        = LLs(ind);
plot(disc_alpha, mll, 'k.', 'MarkerSize', 18);

legend('Numerical MLE curve with xmin=1',...
       'MLE with continuous distr',...
       'MLE with discrete distr',...
       'Location','SouthWest');

p_disc = beta*(x/xmin).^-disc_alpha;

% Show the fits on the original graph
subplot(121);
plot(x,p_cont,'r--');
plot(x,p_disc,'k--','LineWidth',2);
ylim([10^-6 10^0]);
legend('Empirical distr',...
       'MLE fit with continuous distr',...
       'MLE fit with discrete distr',...
       'Location','NorthEast');
```