```
% Dohner HW4
% Question 1 part a and b
% Linear estimate interpolation using gaussian correlation.
%
% code adapted from Dan Rudnick

d=input('Data (2-vector for values at t=[-1 1])? ');
scale=input('e-folding scales? ');
noise=input('Noise? ');
d=d(:);
t=(-5:0.1:5)';
skill=zeros(length(t),length(scale));
x=zeros(size(skill));
skillt=zeros(size(skill));
xt=zeros(size(skill));
for n=1:length(scale)
   % cov = data covariance matrix
   cov=[1+noise (1+(2/scale(n))^2)^-1; (1+(2/scale(n))^2)^-1 1+noise];
   % ct = covariance of data with the signal
   ct=[(1+((t+1)/scale(n)).^2).^-1 (1+((t-1)/scale(n)).^2).^-1];
   skill(:,n)=diag(ct/cov*ct');
   x(:,n)=ct/cov*d;
   ctt=-2/(scale(n).^2)*[t+1 t-1].*ct;
   skillt(:,n)=diag(ctt/cov*ctt')/(2/(scale(n).^2));
   xt(:,n)=ctt/cov*d;
end
figure;
subplot(2,1,1)
plot(t,x),xlabel('t'),ylabel('x');
%legend('L=1' 'L=2' 'L=3' 'L=4' 'L=5' 'L=6' 'L=7' 'L=8' 'L=9' 'L=10');
title('Interpolations between input data points 0.5 and 1 with 0.1 N:S');
subplot(2,1,2)
plot(t,skill),xlabel('t'),ylabel('skill')
figure;
subplot(2,1,1)
plot(t,xt),xlabel('t'),ylabel('dxdt');
subplot(2,1,2)
plot(t,skillt),xlabel('t'),ylabel('skill-dxdt')
```