Julia Dohner
March 16, 2018
SIOC 221B

HW #3

1.

a. See Figures 1 and 2 (top subplots).
b. See Figures 1 and 2 (bottom subplots).
c. See Figures 3 and 4.
Without noise, the skill begins to asymptote at nearly 1 after 6 data points. The growth towards a skill of 1 is not linear, however. Instead, no information is added when using three points instead of two because the third point is added at the mean, thus the skill remains unchanged as the number of data increase from 1 to 2. The skill again increases with the use of 4 data points but drops with 5. This occurs because the fifth point is added at the mean, adding no new information, and pushes (visually) the surrounding data further from the mean, increasing the distance between the points and thus decreasing the skill. After 5 points, however, the skill beings to asymptote at 1. With noise, the skill follows the same pattern with the addition of new data points, but never asymptotes or reaches 1. Instead, the skill slowly increases after 6 data points are used, but reaches a maximum of 0.8649.

2.

a. See Figures 5 and 6.
b. See Figures 7 and 8.
c. Based on the results of the EOF analysis, I think the data for y1 were generated through the superposition of two sine waves with differing frequencies, one of which is split into two phases, each phase being applied to half the data. I think the data for y2 were generated through a random walk procedure because the first EOF shows an oscillatory pattern whereas the second EOF mostly looks to be white noise.

3.

a. See attached MATLAB code. The expression for the gain and the skill are the following:

$$\alpha = \frac{<\tau^* u>}{<\tau^*\tau>} \qquad \rho^2 = \frac{<\tau^* u><u^*\tau>}{<\tau^*\tau><u^*u>}$$

b. The Ekman depth is estimated to be 45.4130 m, and the latitude is estimated to be 90° S.

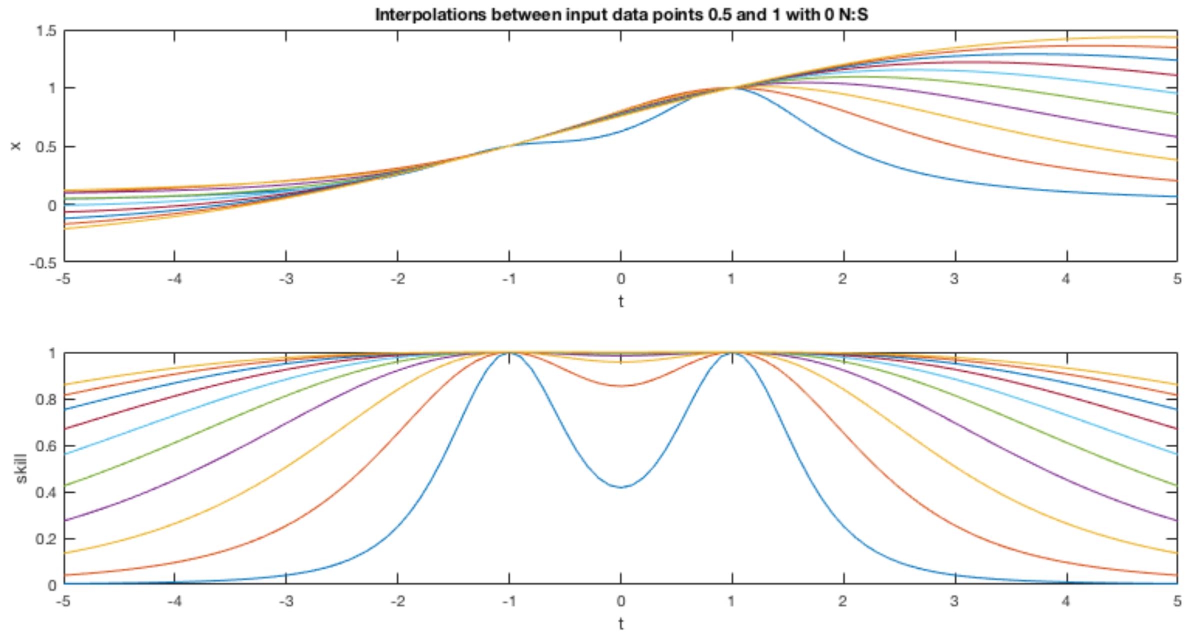

Figure 1. Estimate and skill of variable as a continuous function given data at 0.5 and 1 located at x = -1 and 1 with 0 noise-to-signal ratio.

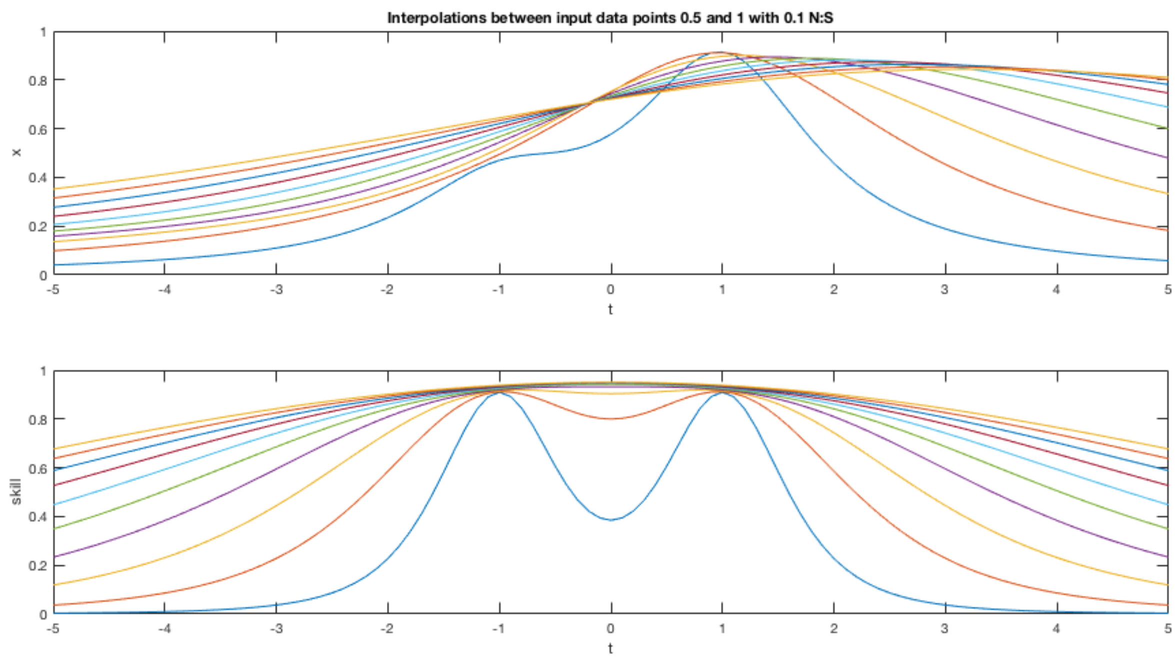

Figure 2. Estimate and skill of variable as a continuous function given data at 0.5 and 1 located at x = -1 and 1 with 0.1 noise-to-signal ratio.

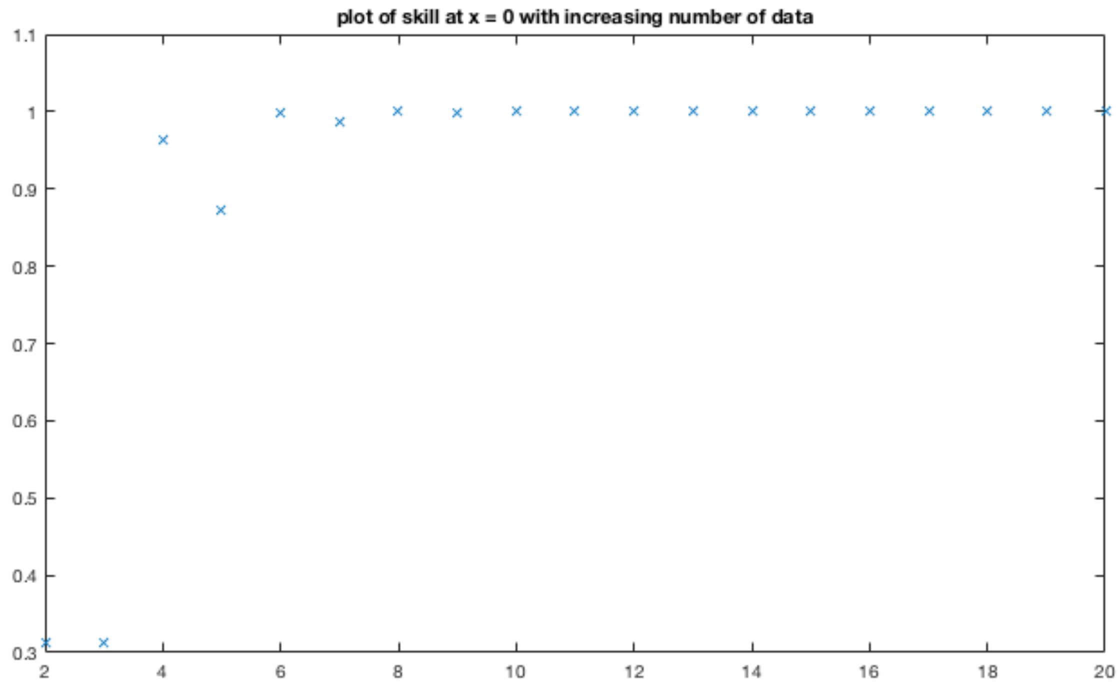

Figure 3. Plot of skill of estimates using increasing number of data points spread evenly between x = -1 and 1 with 0 noise-to-signal ratio.

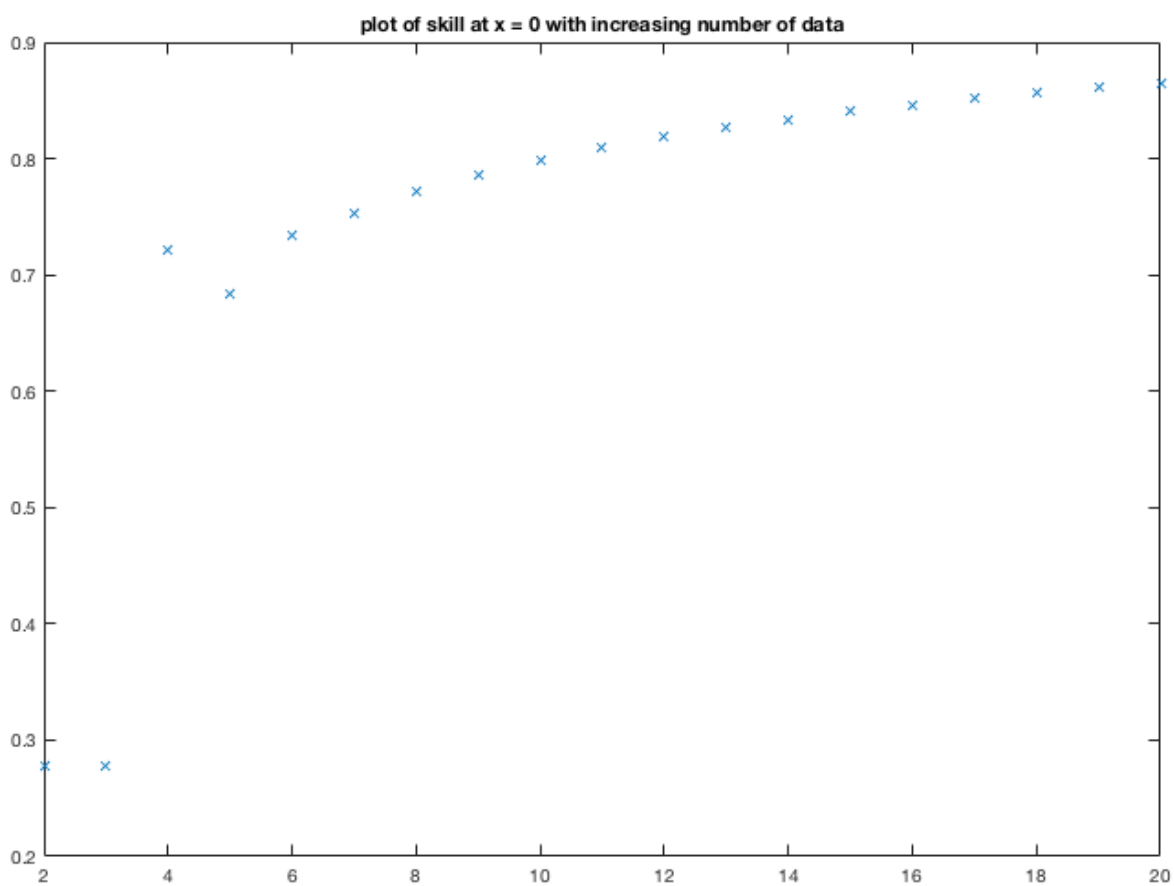

Figure 4. Plot of skill of estimates using increasing number of data points spread evenly between x = -1 and 1 with 0.1 noise-to-signal ratio.

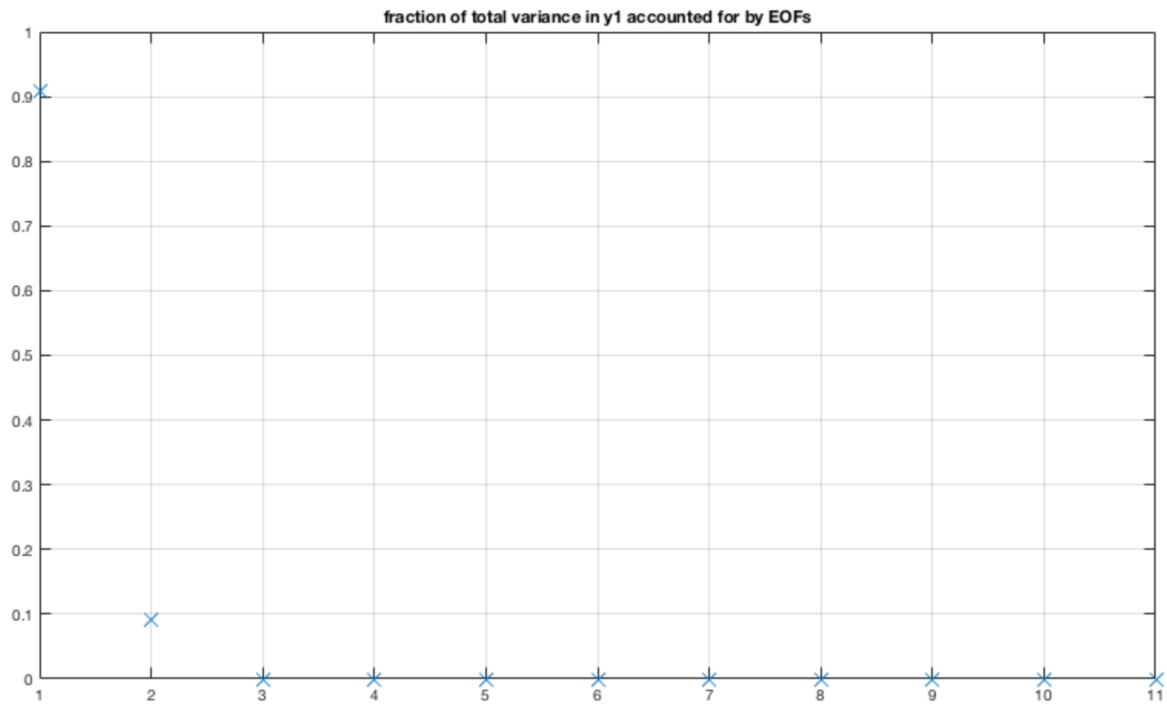

Figure 5. Fraction of total variance in y1 accounted for by each EOF.

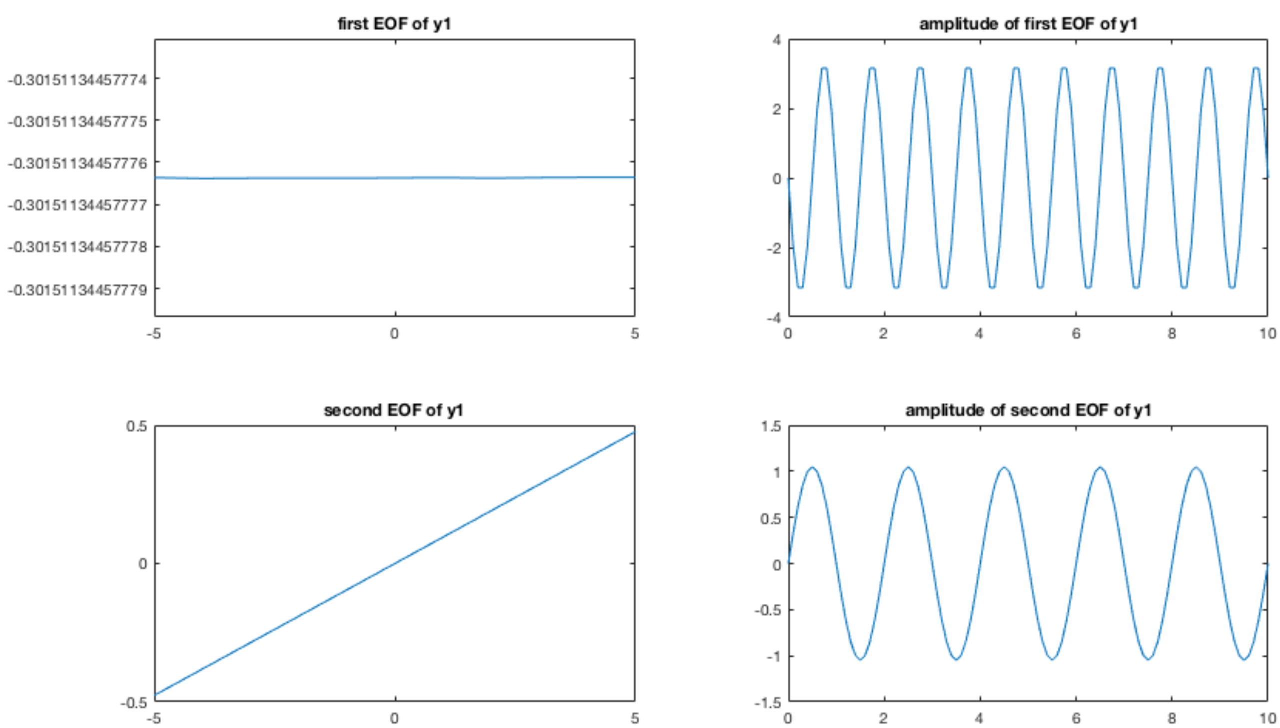

Figure 6. First two EOFs of y1 and their amplitudes.

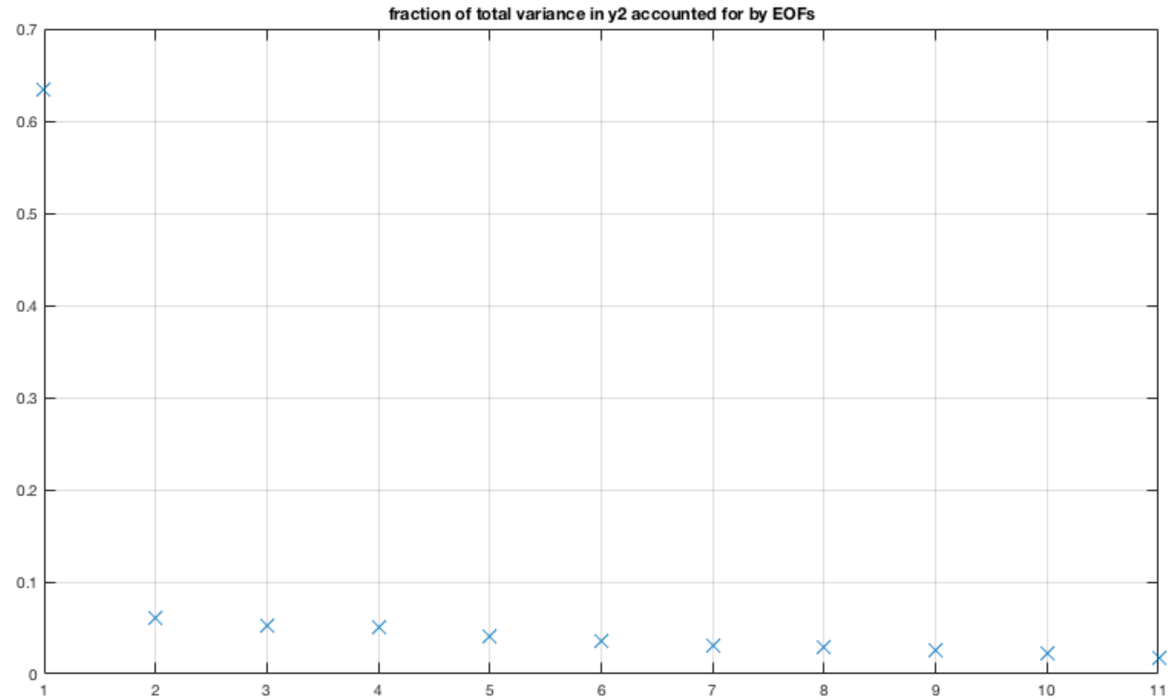

Figure 7. Fraction of total variance in y2 accounted for by each EOF.

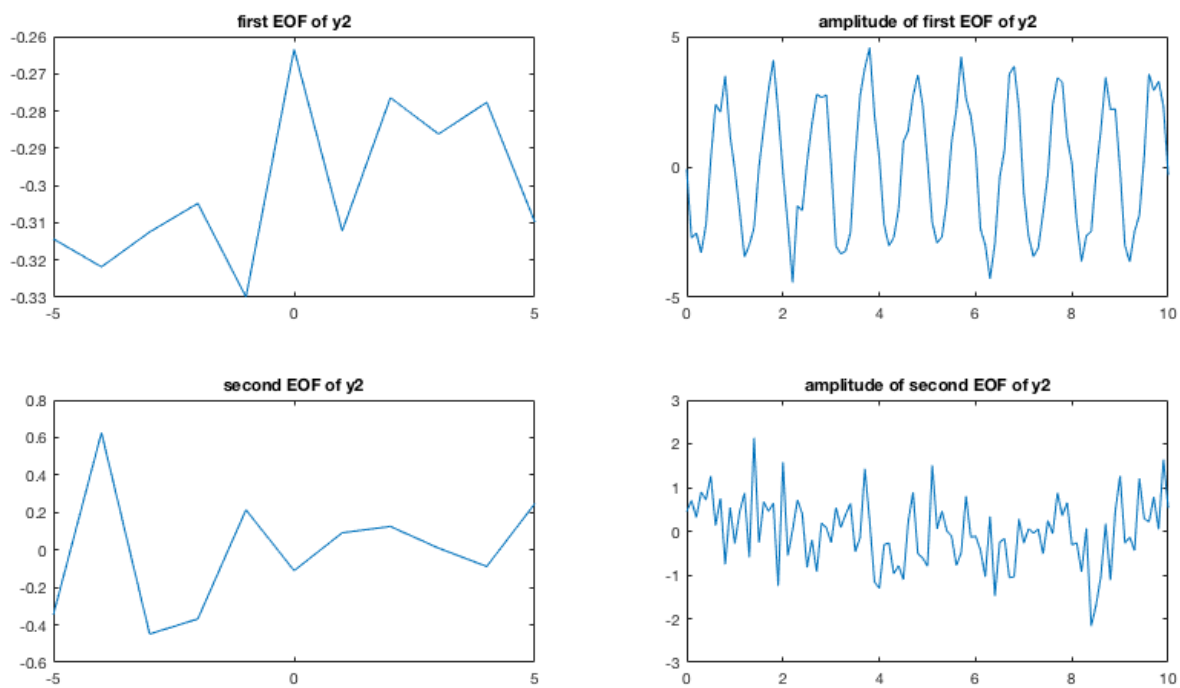

Figure 8. First two EOFs of y1 and their amplitudes.

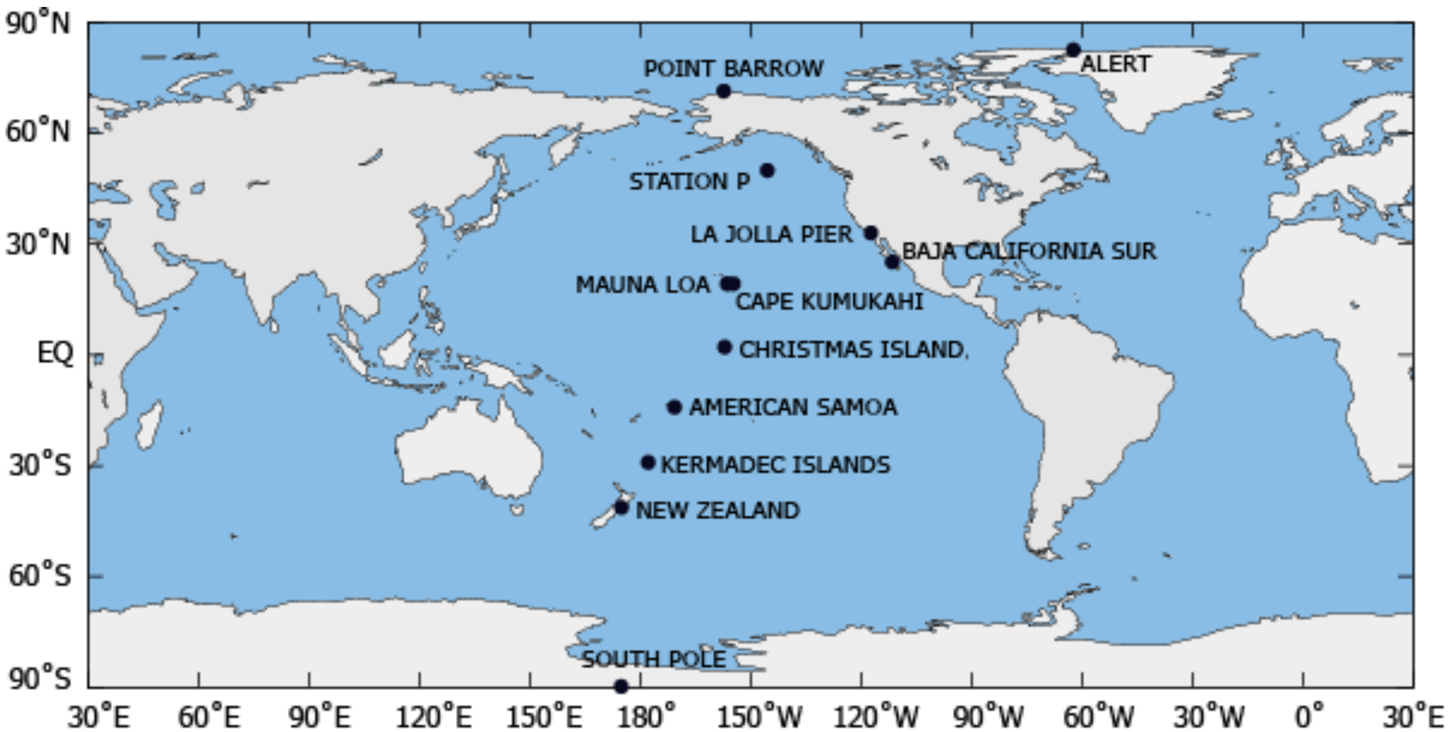

Figure 9. Map of CO_2 measurement stations. Source: http://scrippsco2.ucsd.edu/

Calculations to try with your data:

1. See Figures 10 and 11. I computed EOFs for flask-collected monthly averages of atmospheric CO_2 collected at five different stations: Mauna Loa, La Jolla, Christmas Island, Cape Kumukahi, and Point Barrow. I plotted the first and second EOF's for the data. The first EOF looks to be a long-term linear growth trend with some seasonal cycle, for which all of the station time series are in phase. The second EOF is mostly the seasonal cycle, and it out of phase for some of the stations, possible for the ones that are further south or towards the poles (away from the major concentration of boreal forests at mid-latitudes that strongly influence the seasonal cycle).
2. I supposed a linear relationship between the Mauna Loa CO_2 record and the Point Barrow CO_2 record. The gain of the linear estimate was 1.0041 and the mean-square error was 22.7825.
3. See Figures 14 and 15. I made an objective map of a more or less vertical line at around 155° W, using data from Point Barrow, Station P, Mauna Loa, Christmas Island, American Samoa, New Zealand, and South Pole Observatory, all taken in September of 1981. We'd expect CO_2 to be lower in the northern hemisphere than in the southern hemisphere because September is at the tail end of the summer, meaning that CO_2 levels will be lower. I can see that as I increase the scale, the skill improves between data points. I can also see that increasing the amount of noise pulls the mapped values closer to the mean and away from the data points. The analysis would benefit from more northern hemisphere points.

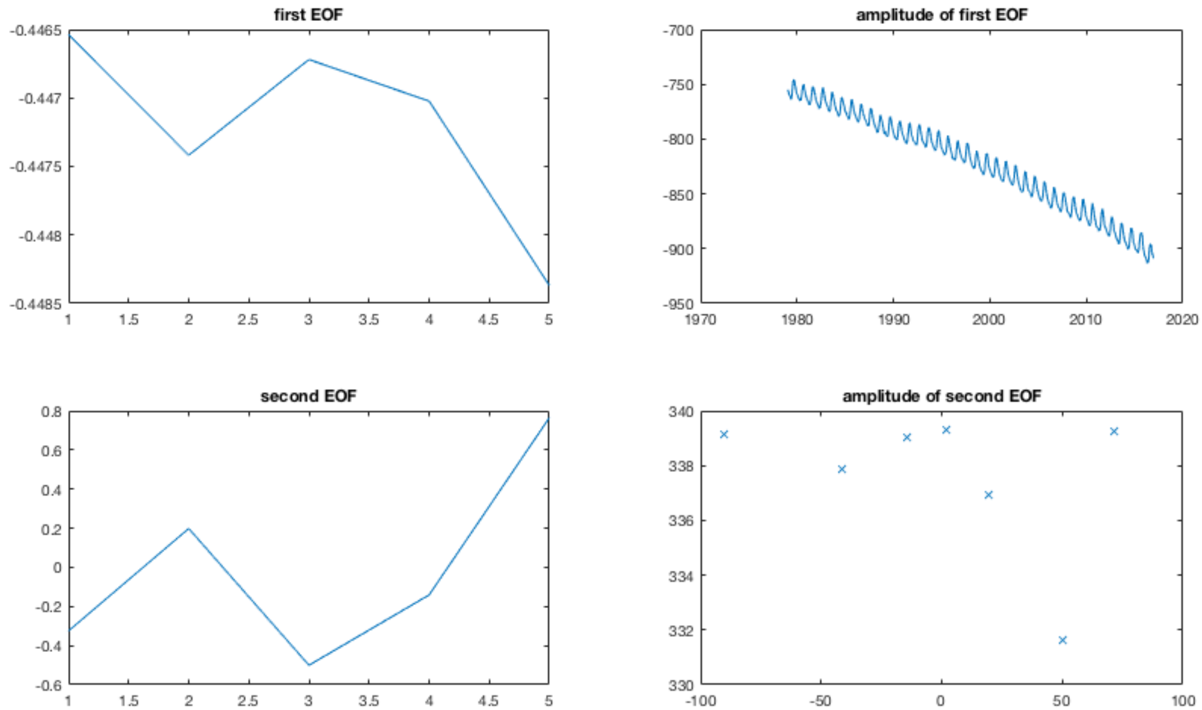

Figure 10. First two EOFs of the 5 CO_2 records and their amplitudes.

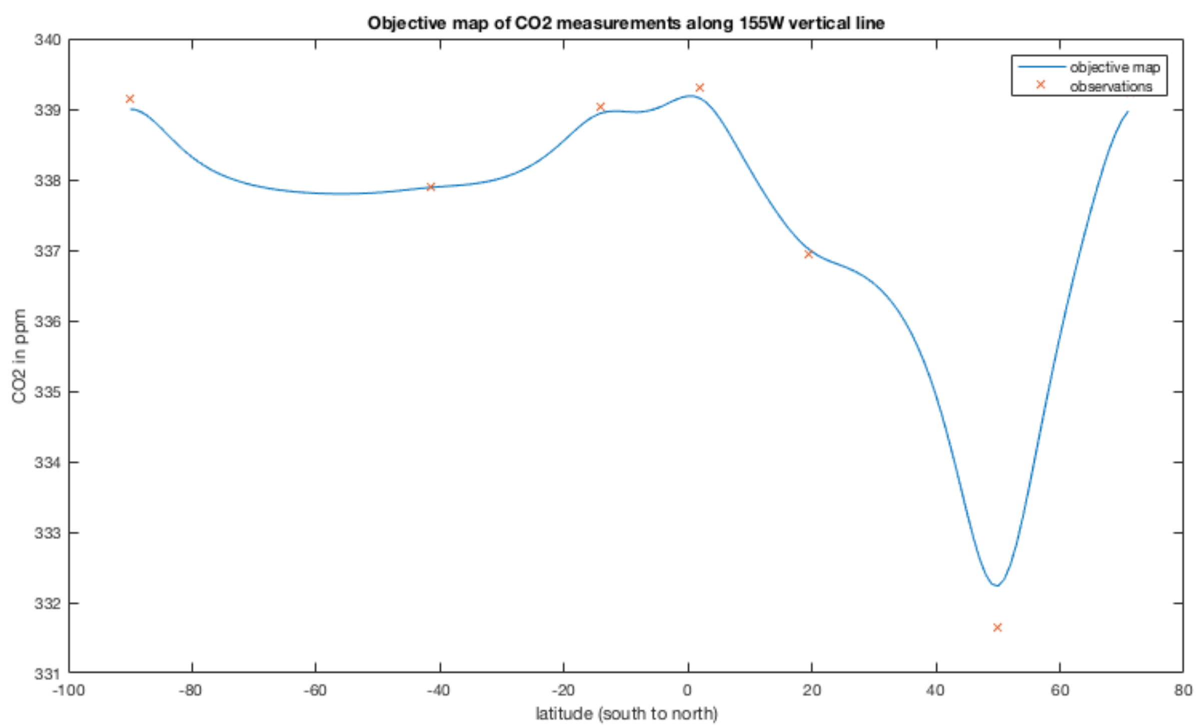

Figure 11. Objective map of CO_2 on north-south line between five stations. Noise-to-signal ratio is 0.1.

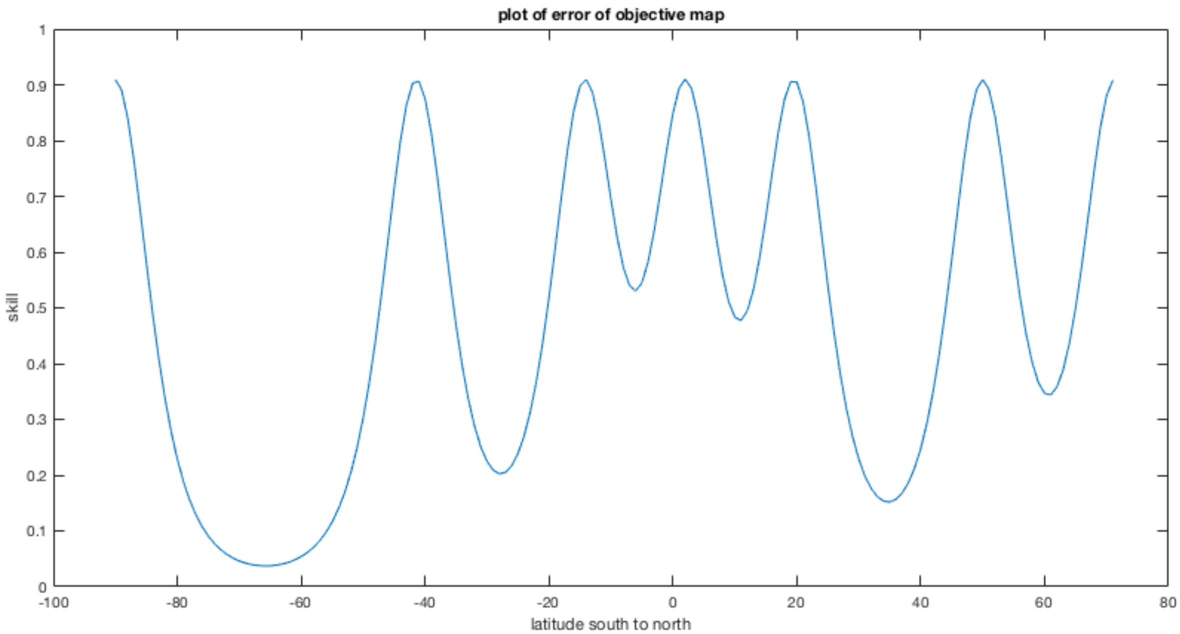

Figure 12. Error of objective map where noise-to-signal ratio is 0.1

```
% 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')
```

```
% HW4 intgauss Question 1 part c
%
% code adapted from Dan Rudnick

clear all; close all

scale=1;
noise=0.1;
t=(-5:0.1:5)';
d = [0.5; 1];
vals = zeros(19,1);


for n=2:20
   t = linspace(scale*-1,scale,n)';
   cov = zeros(n,n);

   % populate covariance matrix
   for i=1:n % rows
       for j=1:n % cols
           cov(i,j) = (1+((t(i)-t(j))/scale)^2)^-1;
       end
   end
   cov = cov+eye(n,n)*noise;

   dx = -(1+((0-t)/scale).^2).^(-2).*(2*(0-t)/(scale^2));% n x 1
   ddx = 2/(scale^2);
   skillt = (dx'*inv(cov)*dx)/(ddx);
   vals(n-1) = skillt;

end

figure
plot(2:20,vals,'x')
title('plot of skill at x = 0 with increasing number of data')
```

```
% eof question of hw #3
% julia dohner
% (question 2)


clear all

% data in matrices y1 and y2 as functions of x and t
% 11 timeseries
% 101 time points in each series

% Perform EOF analysis on y1
% y1 is in form [timeseries timeseries] (as columns)


load eofdata.mat;
%y1 = y1';
y1_dm = detrend(y1, 'constant');
y2_dm = detrend(y2, 'constant');
[U1,S1,V1]=svd(y1_dm,0);
[U2,S2,V2]=svd(y2_dm,0);

figure('Name','y1 data')
subplot(2,1,1)
plot(x,y1(1:11,:))
title('y1 data before removing mean')
subplot(2,1,2)
plot(x,y1_dm(1:11,:)) % removing mean didn't seem to do anything
title('y1 data after removing mean')

figure('Name','y2 data')
subplot(2,1,1)
plot(x,y2(1:11,:))
title('y2 data before removing mean')
subplot(2,1,2)
plot(x,y2_dm(1:11,:))
title('y2 data after removing mean')


% plot the fraction of total variance accounted for by each EOF
figure('Name','y1 fraction of variances')
plot(diag(S1.^2)/trace(S1.^2),'x','markersize',12); % trace is sum of diagnoal
grid;
title('fraction of total variance in y1 accounted for by EOFs')
figure('Name','y2 fraction of variances')
plot(diag(S2.^2)/trace(S2.^2),'x','markersize',12);
title('fraction of total variance in y2 accounted for by EOFs')
grid;

% plot the first two EOFs and amplitudes
figure('Name','y1 EOFs and amplitudes')
subplot(2,2,1)
plot(x,V1(:,1))
title('first EOF of y1')
subplot(2,2,2)
plot(t,U1(:,1)*S1(1,1))
title('amplitude of first EOF of y1')
subplot(2,2,3)
```

```
plot(x,V1(:,2))
title('second EOF of y1')
subplot(2,2,4)
plot(t,U1(:,2)*S1(2,2))
title('amplitude of second EOF of y1')

% amp: plot(t,U(:,2))
%amp2 = V2'*y2_dm;
figure('Name','y2 EOFs and amplitudes')
subplot(2,2,1)
plot(x,V2(:,1))
title('first EOF of y2')
subplot(2,2,2)
plot(t,U2(:,1)*S2(1,1))
title('amplitude of first EOF of y2')
subplot(2,2,3)
plot(x,V2(:,2))
title('second EOF of y2')
subplot(2,2,4)
plot(t,U2(:,2)*S2(2,2))
title('amplitude of second EOF of y2')
```

```
% utau problem
% hw 4 question 3
%
% use page 2 of linear estimation notes

clear all
load utau.mat;

% wind stress tau
% velocity u as a function of depth z

% eastward part real, northward part imaginary

% calculate gain (aka a)
a = zeros(1,length(z));
for i = 1:length(z)
    a(i) = (tau(i)'*u(i))/(tau(i)'*tau(i));
end

% determine skill as a function of depth
skill = zeros(1,length(z));
for i = 1:length(z)
    skill(i) = ((tau'*u(i))*(u(i)'*tau))/((u(i)'*u(i))*(tau'*tau));
end


% estimate Ekman depth and latitude of these data
% De = (2*Av/f)^(1/2) where Av is the eddy viscosity and f is coriolis
% e-folding scale; take the log of the gain?
% use ekman spiral equation at surface?

% Ekman spiral surface equation
rho = 1025;
De = (1+i)*z(2)/log(a(2)/a(1));
depthEk = real(De);

f = (1-i)/a(1)/rho/depthEk;
lat = real(asind(real(f)/2/7.29E-5));
```

```
% hw 4
% Julia Dohner
%
% calcs to try with my data
% using keeling co2 data (flask-sampled monthly averages) at mlo
% code taken from A_SIOC_221/HW9/Dohner_SIOC221A_HW9_monthly.m

clear all; close all;

%% load CO2 data

dataMLO = fopen('monthly_data/monthly_flask_co2_mlo_JLD.txt');
dataLJO = fopen('monthly_data/monthly_flask_co2_ljo_JLD.txt');
dataCHR = fopen('monthly_data/monthly_flask_co2_chr_JLD.txt');
dataKUM = fopen('monthly_data/monthly_flask_co2_kum_JLD.txt');
dataPTB = fopen('monthly_data/monthly_flask_co2_ptb_JLD.txt');




valsMLO = textscan(dataMLO,'%f %f', ...
    'delimiter','\t');
valsLJO = textscan(dataLJO,'%f %f', ...
    'delimiter','\t');
valsCHR = textscan(dataCHR,'%f %f', ...
    'delimiter','\t');
valsKUM = textscan(dataKUM,'%f %f', ...
    'delimiter','\t');
valsPTB = textscan(dataPTB,'%f %f', ...
    'delimiter','\t');


fclose(dataMLO);
fclose(dataLJO);
fclose(dataCHR);
fclose(dataKUM);
fclose(dataPTB);


% format of .txt files is year, co2 value
MLOyear = valsMLO{1};
MLOco2 = valsMLO{2};
LJOyear = valsLJO{1};
LJOco2 = valsLJO{2};
CHRyear = valsCHR{1};
CHRco2 = valsCHR{2};
KUMyear = valsKUM{1};
KUMco2 = valsKUM{2};
PTByear = valsPTB{1};
PTBco2 = valsPTB{2};

% shorten all records to the same length
startYear = KUMyear(1);
chrStart = find(CHRyear == startYear);
CHRyear = CHRyear(chrStart:end);
CHRco2 = CHRco2(chrStart:end);
mloStart = find(MLOyear == startYear);
MLOyear = MLOyear(mloStart:end);
```

```
MLOco2 = MLOco2(mloStart:end);
ljoStart = find(LJOyear == startYear);
LJOyear = LJOyear(ljoStart:end);
LJOco2 = LJOco2(ljoStart:end);
ptbStart = find(PTByear == startYear);
PTByear = PTByear(ptbStart:end);
PTBco2 = PTBco2(ptbStart:end);

% make all of the records end at end of CHR
endInd = length(CHRyear);
MLOyear = MLOyear(1:endInd);
MLOco2 = MLOco2(1:endInd);
LJOyear = LJOyear(1:endInd);
LJOco2 = LJOco2(1:endInd);
KUMyear = KUMyear(1:endInd);
KUMco2 = KUMco2(1:endInd);
PTByear = PTByear(1:endInd);
PTBco2 = PTBco2(1:endInd);

% remove flagged data
for i = 1:length(MLOco2)
    if MLOco2(i) == -99.99
        MLOco2(i) = nan;
    end
end
for i = 1:length(LJOco2)
    if LJOco2(i) == -99.99
        LJOco2(i) = nan;
    end
end
for i = 1:length(CHRco2)
    if CHRco2(i) == -99.99
        CHRco2(i) = nan;
    end
end
for i = 1:length(KUMco2)
    if KUMco2(i) == -99.99
        KUMco2(i) = nan;
    end
end
for i = 1:length(PTBco2)
    if PTBco2(i) == -99.99
        PTBco2(i) = nan;
    end
end

% remove nan's
addpath('/Users/juliadohner/Documents/MATLAB/A_SIOC_221/HW9/Inpaint_nans/Inpaint_nans');
MLOco2 = inpaint_nans(MLOco2);
LJOco2 = inpaint_nans(LJOco2);
CHRco2 = inpaint_nans(CHRco2);
KUMco2 = inpaint_nans(KUMco2);
PTBco2 = inpaint_nans(PTBco2);


% plot timeseries
figure('name','Atmospheric CO2 Timeseries');
plot(MLOyear,MLOco2, '.-',LJOyear,LJOco2,'.-',CHRyear,CHRco2,'.-',...
```

```
    KUMyear,KUMco2,'.-',PTByear,PTBco2,'.-');
xlabel('\fontsize{14}year')
ylabel('\fontsize{14}ppm')
title('\fontsize{16}MLO Atmospheric CO2 Record')
legend('\fontsize{12}Mauna Loa','La Jolla','Christmas Island',...
    'Cape Kumukahi','Point Barrow','location','northwest');

%% calculate EOF of data
% get a bunch of different timeseries from around the world

% organize timeseries into workable matrix
Y = [MLOco2, LJOco2, CHRco2, KUMco2, PTBco2];
[U,S,V] = svd(Y,0);

x = [1:5];
t = MLOyear;
figure('Name','EOFs and amplitudes of CO2 records')
subplot(2,2,1)
plot(x,V(:,1))
title('first EOF')
subplot(2,2,2)
plot(t,U(:,1)*S(1,1))
title('amplitude of first EOF')
subplot(2,2,3)
plot(x,V(:,2))
title('second EOF')
subplot(2,2,4)
plot(t,U(:,2)*S(2,2))
title('amplitude of second EOF')


%% suppose linear relationship

% looking for linear relationship between MLO and PTB


%gain = (MLOco2.*PTBco2)/(MLOco2.^2);
gain = (MLOco2'*PTBco2)./(MLOco2'*MLOco2);
PTBcalc = gain*MLOco2;

MSE = ((PTBco2'*PTBco2) - (PTBcalc'*PTBcalc))./length(PTBco2);

%% create objective map

% can do this just from a single timeseries
dataSTP = fopen('monthly_data/monthly_flask_co2_stp_JLD.txt');
dataSAM = fopen('monthly_data/monthly_flask_co2_sam_JLD.txt');
dataKER = fopen('monthly_data/monthly_flask_co2_ker_JLD.txt');
dataNZD = fopen('monthly_data/monthly_flask_co2_nzd_JLD.txt');
dataSPO = fopen('monthly_data/monthly_flask_co2_spo_JLD.txt');

valsSTP = textscan(dataSTP, '%f %f', ...
    'delimiter','\t');
valsSAM = textscan(dataSAM, '%f %f', ...
    'delimiter','\t');
valsKER = textscan(dataKER, '%f %f', ...
    'delimiter','\t');
valsNZD = textscan(dataNZD, '%f %f', ...
```

```
    'delimiter','\t');
valsSPO = textscan(dataSPO,'%f %f', ...
    'delimiter','\t');

fclose(dataSTP);
fclose(dataSAM);
fclose(dataKER);
fclose(dataNZD);
fclose(dataSPO);

STPyear = valsSTP{1};
STPco2 = valsSTP{2};
SAMyear = valsSAM{1};
SAMco2 = valsSAM{2};
KERyear = valsKER{1};
KERco2 = valsKER{2};
NZDyear = valsNZD{1};
NZDco2 = valsNZD{2};
SPOyear = valsSPO{1};
SPOco2 = valsSPO{2};


% remove flagged data
for i = 1:length(NZDco2)
    if NZDco2(i) == -99.99
        NZDco2(i) = nan;
    end
end
for i = 1:length(KERco2)
    if KERco2(i) == -99.99
        KERco2(i) = nan;
    end
end
for i = 1:length(SAMco2)
    if SAMco2(i) == -99.99
        SAMco2(i) = nan;
    end
end
for i = 1:length(STPco2)
    if STPco2(i) == -99.99
        STPco2(i) = nan;
    end
end

% need to interpl for STP because NaN at time point want to use
STPco2 = inpaint_nans(STPco2);

% find year = 1.981041100000000e+03
% this is the earliest year for SAM, overlaps with latest yr for STP
useYear = STPyear(end-2);%SAMyear(2);

datumPTB = PTBco2(find(PTByear == useYear));
datumSTP = STPco2(find(STPyear == useYear));
datumMLO = MLOco2(find(MLOyear == useYear));
datumCHR = CHRco2(find(CHRyear == useYear));
datumSAM = SAMco2(find(SAMyear == useYear));
datumNZD = NZDco2(find(NZDyear == useYear));
datumSPO = SPOco2(find(SPOyear == useYear));
```

```
d=[datumPTB, datumSTP, datumMLO, datumCHR, datumSAM, datumNZD, datumSPO];
avgD = mean(d);
d = d-avgD;
scale=10;
noise=0.1; % assuming 0 noise, physically unrealistic
d=d(:);
lats = [71.3, 50.0, 19.5, 2.0, -14.2, -41.4, -90];
t = (-90:1:71.3);
skill=zeros(length(t),length(scale));
x=zeros(size(skill));
skillt=zeros(size(skill));
xt=zeros(size(skill));
numData = length(d);

cov = zeros(numData,numData);
% populate covariance matrix
for i=1:numData % rows
   for j=1:numData % cols
       cov(i,j) = (1+((lats(i)-lats(j))/scale)^2)^-1;
   end
end
cov = cov+eye(numData,numData)*noise;

% ct = covariance of data with the signal
ct = zeros(length(t),numData);
for i=1:numData % rows
   for j=1:length(t) % cols
       ct(j,i) = (1+((lats(i)-t(j))/scale)^2)^-1;
   end
end

skill=diag(ct/cov*ct');
x=ct/cov*d;
x = x+avgD;


figure
plot(t,x)
hold on
plot(lats,d+avgD,'x');
title('Objective map of CO2 measurements along 155W vertical line')
xlabel('latitude (south to north)')
ylabel('CO2 in ppm')
legend('objective map','observations')

figure
plot(t,skill)
title('plot of error of objective map')
```