```
% 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')
```