

```
% file SIOC 221A HW 2
%
% author Julia Dohner, with help from Margaret Lindeman on subsampling
%
% due date October 12, 2017

clear all; close all;

numYears = 2017-2005 + 1;

%% plotting the automated salinity record

% create empty arrays to hold time and temp data
timeAuto = [];
salinityAuto = [];

% loop through each year starting with 2005 to retrieve data
for i = 1:numYears
    binomialDist = (i-1) + 2005;% get value for each year starting with first year
    timeAuto = [timeAuto; ncread(strcat('http://sccoos.org/thredds/dodsC/autoss/scripps_pier-', num2str(binomialDist), '.nc'), 'time')];
    salinityAuto = [salinityAuto; ncread(strcat('http://sccoos.org/thredds/dodsC/autoss/scripps_pier-', num2str(binomialDist), '.nc'), 'salinity')];
end

% remove bad data using the flagged data from .nc file
% create empty arrays to hold time and temp data
salinity_flagPrimary_Auto = [];

% retrieve flagged data
for i = 1:numYears
    binomialDist = (i-1) + 2005;% get value for each year starting with first year
    salinity_flagPrimary_Auto = [salinity_flagPrimary_Auto; ncread(strcat('http://sccoos.org/thredds/dodsC/autoss/scripps_pier-', num2str(binomialDist), '.nc'), 'salinity_flagPrimary')];
end

%looping through to remove bad data from salinity record
for i = 1:length(salinityAuto)
    if salinity_flagPrimary_Auto(i) ~= 1
        salinityAuto(i) = nan;
    end
end

% plot the time series
date0=datenum(1970,1,1); % give reference date (first date)
time = double(timeAuto/24/3600+date0);
figure('name','Scripps_Pier_Salinity_2005-2017_Automated');
plot(time, salinityAuto,'LineWidth',1);

% label the x-axis in months
datetick('x','yyyy');
set(gca,'FontSize',16);
title('Scripps Pier Salinity - Automated Record');
xlabel('Date');
datetick('x','mm/dd/yy','keeplimits')
```

```
ylabel('Practical Salinity Units');

% mean salinity
meanSalinityAuto = nanmean(salinityAuto);
stdSalinityAuto = nanstd(salinityAuto);

%% plotting the manual salinity record

filein = 'SI0_SALT_1916-201410.txt';
fileID = fopen(filein);
%read in header info
headerInfo = textscan(fileID, '%s', 27, 'delimiter', '\n');
dataInfo = textscan(fileID, '%s', 9, 'delimiter', '\t');
% read in the data
matchingTimes = textscan(fileID, '%f %f %f %f %f %f %f %f %f', 'delimiter', '\t \t \t \t \t \t \t \t \t \t');

%extract the data from the cell matrix
yearDataManual = matchingTimes{1};
monthDataManual = matchingTimes{2};
dayDataManual = matchingTimes{3};
timeDataManual = matchingTimes{4};
timeFlagDataManual = matchingTimes{5};
salinityDataManual = matchingTimes{6};
salinityFlagDataManual = matchingTimes{7};
bottleSalDataManual = matchingTimes{8};
bottleSalFlagDataManual = matchingTimes{9};

%looping through to remove bad data from salinity record
for i = 1:length(yearDataManual)
% ignoring flags for time
%    if timeFlagData(i) ~= 0
%        salinityData(i) = nan;
    if salinityFlagDataManual(i) ~= 0
        salinityDataManual(i) = nan;
    elseif bottleSalFlagDataManual(i) ~= 0
        bottleSalDataManual(i) = nan;
    elseif salinityDataManual(i) == 720
        salinityDataManual(i) = nan;
    end
end

% convert all nan times to noon
for i = 1:length(timeDataManual)
    if isnan(timeDataManual(i)) == 1
        timeDataManual(i) = 1200;
    end
end

% convert time data to hours and minutes
time_hour = floor(timeDataManual/100);
time_minute = timeDataManual-time_hour*100;
time_second = zeros(length(yearDataManual),1);

%turn the year, mo, day, time into a MATLAB date
timeManual = datenum(yearDataManual, monthDataManual, dayDataManual, time_hour,
time_minute, time_second);
```

```
%plot time series
figure('name','Scripps_Pier_Salinity_1916-2004_Manual');
plot(timeManual,salinityDataManual,'-')
set(gca,'FontSize',16);
t1 = datenum('22-august-1916');
t2 = datenum('31-october-2014');
xlim([t1 t2]);
%label the plot
datetick('x','mm/dd/yy','keeplimits')
xlabel('Date')
title('Scripps Pier Salinity - Manual Record')
ylabel('Practical Salinity Units');

% mean salinity
meanSalinityManual = nanmean(salinityDataManual);
stdSalinityManual = nanstd(salinityDataManual);

%% subsampling to compare manual and auto

% choosing only the automated data taken at the same time as the manual
% data

% creating new vector of manual time data without seconds data
datevecManual = datevec(timeManual);
datevecManual_trunc = [datevecManual(:,1:5), zeros(length(datevecManual),1)];
datevecManual_rounded = datenum(datevecManual_trunc);

% creating new vector of automated time data without seconds data
timeSubsampAuto = double(timeAuto)/24/3600+date0;
datevecAuto = datevec(timeSubsampAuto);
datevecAuto_trunc = [datevecAuto(:,1:5), zeros(length(datevecAuto),1)];
datevecAuto_rounded = datenum(datevecAuto_trunc);

% find times in automated and manual records that match
[matchingTimes, indexAuto, indexManual] = intersect(datevecAuto_rounded,
datevecManual_rounded);

meanSubAuto = nanmean(salinityAuto(indexAuto));
stdSubAuto = nanstd(salinityAuto(indexAuto));

meanSubManual = nanmean(salinityDataManual(indexManual));
stdSubManual = nanstd(salinityDataManual(indexManual));

%% theoretical PDFs

x = meanSalinityAuto-(4*stdSalinityAuto):0.001:meanSalinityAuto+(4*stdSalinityAuto);

% using preset MATLAB distributions:

% gaussian preset
gaussianY = pdf('Normal', x, meanSalinityAuto, stdSalinityAuto);

% uniform preset
% solved for upper and lower bounds in wolfram alpha using:
% 1/12*(upper-lower)^2 = std^2
% 0.5*(upper+lower) = mean
```

```
pdUniform = makedist('Uniform','lower', 32.6138, 'upper', 34.058);
uniformY = pdf(pdUniform,x);

% fake datasets:

% fake gaussian dataset
gaussianDist = normrnd(meanSalinityAuto,stdSalinityAuto,[1,3337]);

% creating fake uniform distribution dataset
% subtract 0.5 to center mean at 0
uniformDist = rand(3337,1)-0.5;%+ meanSalinityAuto;1.41198*
uniformDist = uniformDist*(stdSalinityAuto/std(uniformDist)),% scale the standard deviation
uniformDist = uniformDist + (meanSalinityAuto - mean(uniformDist));

% creating fake bimodal distribution dataset
binomialDist = zeros(1,3337);
for i = 1668:3337
    binomialDist(i) = 10;
end
%scale matrix n
binomialDist = binomialDist-5;
binomialDist = binomialDist*(stdSalinityAuto/std(binomialDist)),% scale the standard deviation
binomialDist = binomialDist + (meanSalinityAuto - mean(binomialDist)),%scale the mean

figure
subplot(5,1,1);
plot(x,gaussianY);
title('Theoretical Probability Density Function Plots');
legend('MATLAB Gaussian PDF');
xlabel('Salinity (psu)');
ylabel('Probability');
subplot(5,1,2);
plot(x,uniformY);
legend('MATLAB Uniform PDF');
xlabel('Salinity (psu)');
ylabel('Probability');
subplot(5,1,3);
EDGES = 31.5:0.001:35;
histogram(gaussianDist,EDGES,'Normalization','pdf');
legend('Fake Data Gaussian PDF');
xlabel('Salinity (psu)');
ylabel('Probability');
subplot(5,1,4)
histogram(uniformDist,EDGES,'Normalization','pdf');
legend('Fake Data Uniform PDF');
xlabel('Salinity (psu)');
ylabel('Probability');
subplot(5,1,5)
histogram(binomialDist, EDGES,'Normalization','pdf');
legend('Fake Data Binomial PDF');
xlabel('Salinity (psu)');
ylabel('Probability');
```

```
%% empirical probability density functions

figure('name','PDF_Scripps_Pier_Salinity');
subplot(2,1,1)
histogram(salinityAuto,'Normalization','pdf');

set(gca,'FontSize',16);
title('Probability Density Function of Salinity for Automated Data');
xlabel('Salinity (psu)','FontSize',16);
ylabel('probability', 'FontSize',16);

subplot(2,1,2)
EDGES = 30:0.1:35;
histogram(salinityDataManual,EDGES,'Normalization','pdf'); %indicate how many bins

minManual = nanmin(salinityDataManual);
maxManual = nanmax(salinityDataManual);

set(gca,'FontSize',16);
title('Probability Density Function of Salinity for Manual Data');
% xlim([29.64, 34.8600]) % MATLAB wouldn't take variables here but they're the min and max
% of manual salinity values
xlabel('Salinity (psu)','FontSize',16);
ylabel('probability', 'FontSize',16);

%% compare PDFs

% calculate cdf
% autoCDF = cdfplot(salinityAuto);
% manuCDF = cdfplot(salinityDataManual);
%
% h = kstest2(autoCDF,manuCDF);

stderrorAuto = stdSalinityAuto/sqrt(length(salinityAuto)),% 3.7e-04
stderrorManual = stdSalinityManual/sqrt(length(salinityDataManual)),%9.7e-4
```