SIOC 221A Homework 2 **Julia Dohner**
Prof. Sarah Gille Due Date: October 12th, 2017

1. Visual evaluation.

It seems as though the automatic data has a lot of readings below 32, whereas the manual data does not, likely due to biofouling, which lowers the conductivity readings. The automatic data is now being collected every four minutes, whereas the manual data is collected in the late morning, usually between 9 am and noon, but in some cases as late 2 pm. The manual record goes back as far as 1916, whereas the automated record began in 2005.

2. Means.

The mean salinity from the automated record is 33.34 +/- 0.42, while the mean salinity from the manual record is 33.58 +/- 0.18. The mean salinity is consistent within error bars.

3. Variance.

The standard deviation is 0.18 for the manual record and 0.42 for the automated record. Based on the plots, I expected the standard deviation for the automated record to be larger (also due to issues such as biofouling). I subsampled the data and compared only the data in the automated and manual records that were taken at the exact same time. The mean of the subsampled manual record is 33.59 with a standard deviation of 0.15.The mean of the subsampled automated record is 33.27 with a standard deviation of 0.50. The means appear to be within the margins of error, mostly because the standard deviation of the automated data is so wide. Subsampling seems to have decreased the statistical similarities between the automated and manual records.

The subsampling technique was not perfect because it only used 630 datapoints of each of the records. I used only the data where the measurements were taken at the exact same time (hours and minutes). I could have included more data points if I included data from the automated record that was close to when the manual data was taken (for example, automated data from 12:28 pm and manual data from 12:30 pm), assuming that the recorded time from the manual data might not be precise. That being said, I decided it best not to make the assumption that the manual time data was faulty and chose to compare only those that were taken at the exact same time (to the best of our knowledge).

4. Theoretical pdfs.

The pdfs for the observed mean and variance of the automated data are displayed on the following pages.

5. Empirical probability density functions.

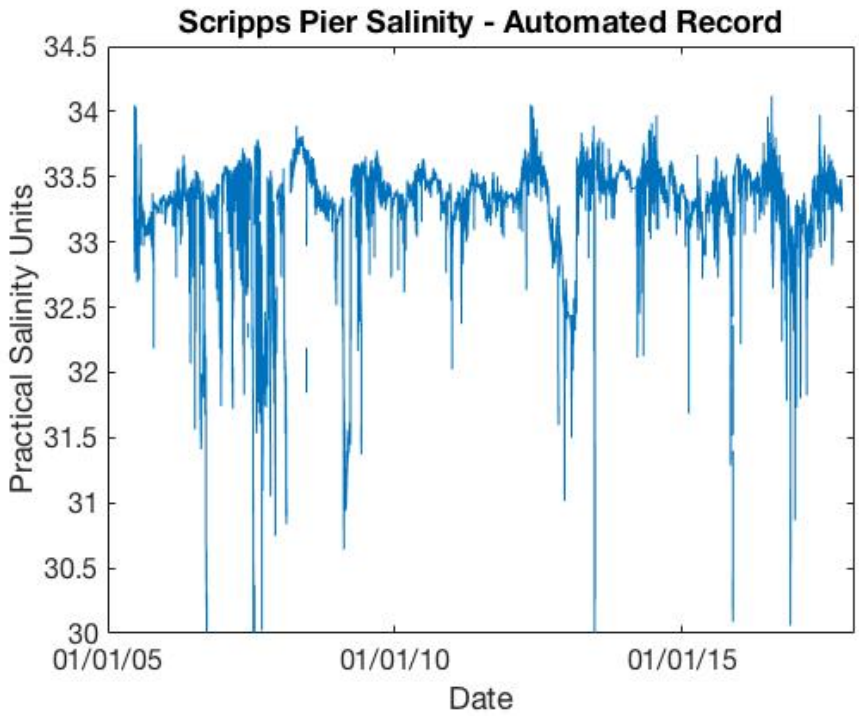
The empirical pdfs for the manual and automated data are displayed on the following pages.

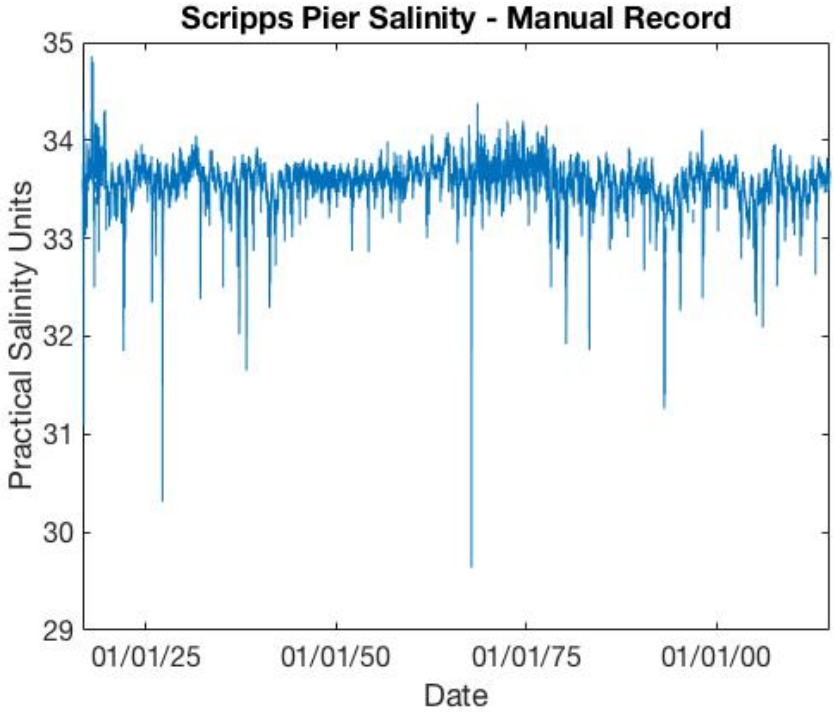
6. Compare pdfs.

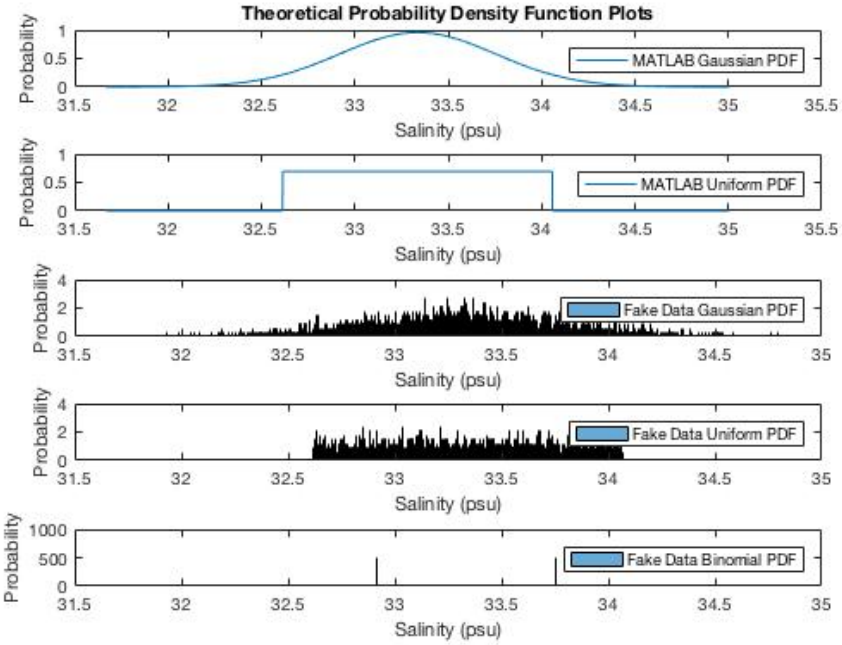
I attempted the Kolmogorov-Smirnov test, but ran into problems and instead chose to calculate the standard errors of both datasets. The standard error of the mean of the automated record is 3.7e-4, while the standard error of the mean of the manual dataset is 9.7e-4. The standard error for both seems very small. The means do not overlap within the standard error, meaning that the means are not the same within error.

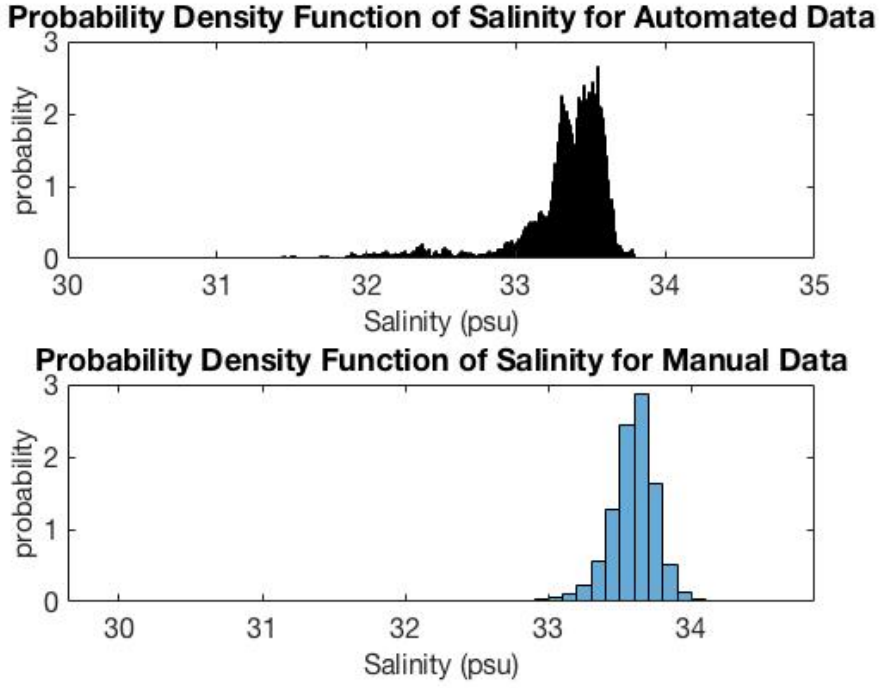
7. Summary.

After evaluating both datasets, I realize that the manual record is valuable because it better constraints the variability in the automated salinity records. Without the manual record, the swings in the salinity values on the automated record could be considered to be a reflection of oceanic variability, whereas they are instead likely due to issues associated with automated sampling, such as biofouling. On the other hand, the automated record is important because it is a much higher resolution than the manual record and can resolve variability over the course of a day (whereas the manual measurements are taken only once per day). Having both records is important for validating each of the measurements, as well as for continuing the extensive (over 100 years long!) time series.


Scripps Pier Salinity - Automated Record
34.5
34
33.5
33
32.5
32
31.5
31
30.5
30
Practical Salinity Units
01/01/05
01/01/10
01/01/15
Date


Scripps Pier Salinity - Manual Record
35
34
33
32
31
30
29
Practical Salinity Units
01/01/25
01/01/50
01/01/75
01/01/00
Date


Theoretical Probability Density Function Plots
Probability
MATLAB Gaussian PDF
MATLAB Uniform PDF
Fake Data Gaussian PDF
Fake Data Uniform PDF
Fake Data Binomial PDF
Salinity (psu)


Probability Density Function of Salinity for Automated Data
probability
Salinity (psu)
Probability Density Function of Salinity for Manual Data
probability
Salinity (psu)

```
% 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 = 'SIO_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');

%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
```