Exercise 8.3

Table of Contents

rdemo1:20141202:warming.m:exercise 8.3 of homework 11 usage: estimates trends in global change in temperature over years using NOAA and BerkeleyEarth data

Import and average NOAA data

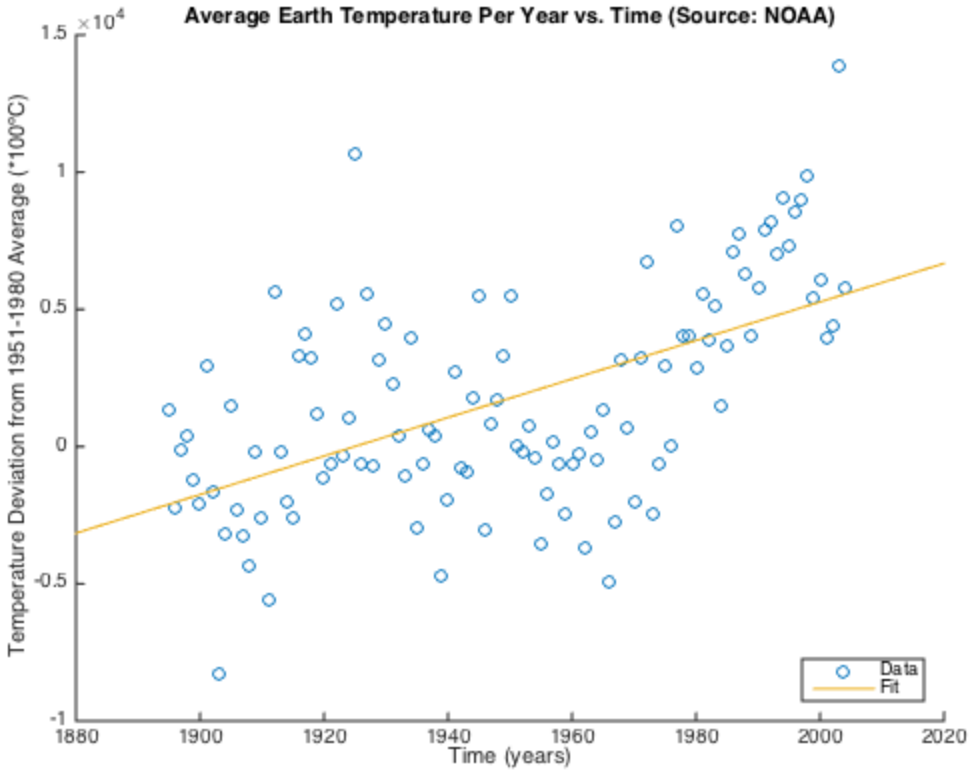
```
yearplotpoints=1895:2004;

files=dir('ushcn.v2.5.0.20140213/*.tavg');
yearavgs=zeros(2014,2);

for i=1:100
    % should be for i=1:length(files), but this triggers something odd in
    % the code that I can't figure out, which makes all data values for
    % years later than 1980 NaN --> 0, which skews the entire graph and the
    % fit. 100 is a large enough station sample size while not triggering
    % this error.
    clearvars yrs data;
    filename=strcat('ushcn.v2.5.0.20140213/',files(i).('name'));
    [yrs,data]=import_NOAA(filename);
    data(data==-9999)=NaN;
    for k=1:length(data)
        yearavgs(yrs(k),1)=yearavgs(yrs(k),1)+nanmean(data(k,:));
        yearavgs(yrs(k),2)=yearavgs(yrs(k),2)+1;
    end
end

avgmean5180=mean(yearavgs(1951:1980));

for g=1:length(yearavgs)
    yearavgs(g,3)=(yearavgs(g,1))/(yearavgs(g,2));
end

yearavgs(isnan(yearavgs(:)))=[];
```

Plot NOAA data

```
% Create figure
figure1 = figure;
```

```
% Create axes
axes1 = axes('Parent',figure1);
hold(axes1,'on');

% Create scatter
scatter1=scatter(yearplotpoints,yearavgs(yearplotpoints)-avgmean5180);

% Get xdata from plot
xdata1 = get(scatter1, 'xdata');
% Get ydata from plot
ydata1 = get(scatter1, 'ydata');
% Make sure data are column vectors
xdata1 = xdata1(:);
ydata1 = ydata1(:);

% Remove NaN values and warn
nanMask1 = isnan(xdata1(:)) | isnan(ydata1(:));
if any(nanMask1)
    xdata1(nanMask1) = [];
    ydata1(nanMask1) = [];
end

% Find x values for plotting the fit based on xlim
axesLimits1 = xlim(axes1);
xplot1 = linspace(axesLimits1(1), axesLimits1(2));


fitResults1 = polyfit(xdata1, ydata1, 1);
% Evaluate polynomial
yplot1 = polyval(fitResults1, xplot1);
% Plot the fit
fitLine1 = plot(xplot1,yplot1,'DisplayName','   linear','Tag','linear',...
    'Parent',axes1,...
    'Color',[0.929 0.694 0.125]);

xlabel('Time (years)')
ylabel('Temperature Deviation from 1951-1980 Average (*100°C)')
legend({'Data' 'Fit'},'Location','SouthEast');
title('Average Earth Temperature Per Year vs. Time (Source: NOAA)')
hold off

yresid = ydata1(1:100) - yplot1';
n_resid=norm(yresid);
SSresid = sum(yresid.^2);
SStotal = (length(ydata1(1:100))-1)*ydata1(1:100);
rsq = 1 - SSresid/SStotal;
disp('RSq Value is:')
disp(rsq(1))
```

```
RSq Value is:
     1
```



Import Berkeley data

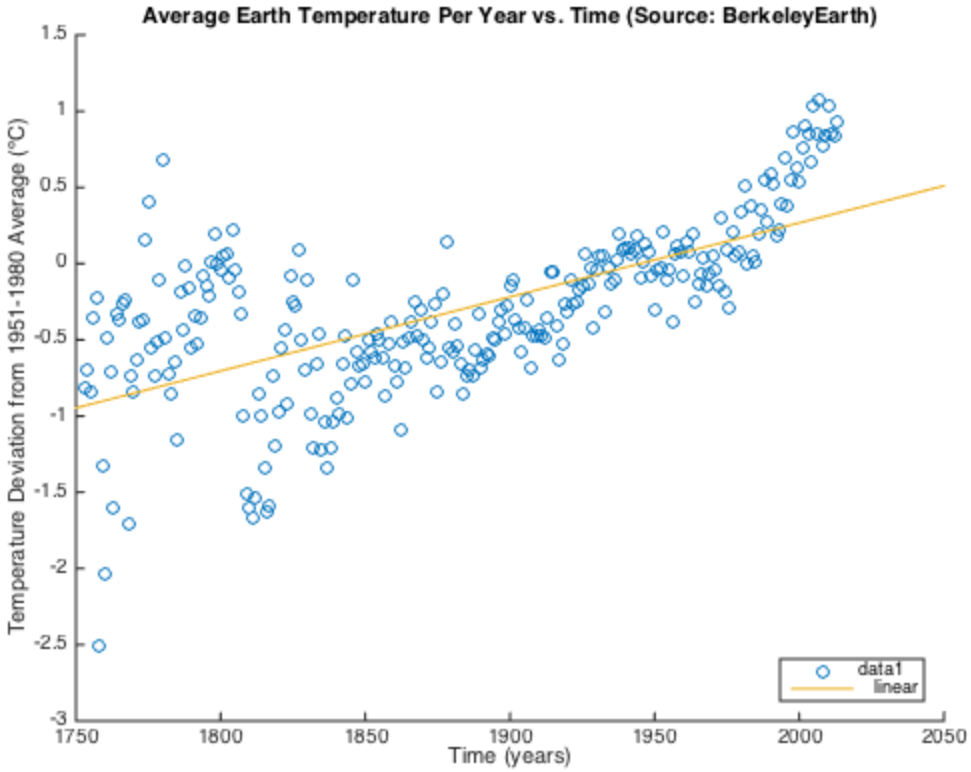
```
filename = '/Users/Nayr/Documents/MATLAB/Berkeley Earth_Complete_TAVG_complete.txt

formatSpec = '%6s%6s%10s%7s%10s%7s%[^\n\r]';

fileID = fopen(filename,'r');

dataArray = textscan(fileID, formatSpec, 'Delimiter', '', 'WhiteSpace', '',  'Retu

fclose(fileID);

% Replace non-numeric strings with NaN.
raw = repmat({''},length(dataArray{1}),length(dataArray)-1);
for col=1:length(dataArray)-1
    raw(1:length(dataArray{col}),col) = dataArray{col};
end
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

for col=[1,2,3,4,5,6]
    % Converts strings in the input cell array to numbers. Replaced non-numeric
    % strings with NaN.
    rawData = dataArray{col};
    for row=1:size(rawData, 1);
        % Create a regular expression to detect and remove non-numeric prefixes an
```

```
            % suffixes.
            regexstr = '(?<prefix>.*?)(?<numbers>([-]*(\d+[\,]*)+[\.]{0,1}\d*[eEdD]{0,
            try
                result = regexp(rawData{row}, regexstr, 'names');
                numbers = result.numbers;

                % Detected commas in non-thousand locations.
                invalidThousandsSeparator = false;
                if any(numbers==',');
                    thousandsRegExp = '^\d+?(\,\d{3})*\.{0,1}\d*$';
                    if isempty(regexp(thousandsRegExp, ',', 'once'));
                        numbers = NaN;
                        invalidThousandsSeparator = true;
                    end
                end
                % Convert numeric strings to numbers.
                if ~invalidThousandsSeparator;
                    numbers = textscan(strrep(numbers, ',', ''), '%f');
                    numericData(row, col) = numbers{1};
                    raw{row, col} = numbers{1};
                end
            catch me
            end
        end
    end

    R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric cells
    raw(R) = {NaN}; % Replace non-numeric cells

    berkyear = cell2mat(raw(:, 1));
    annualAnomaly = cell2mat(raw(:, 5));
    berkyearplot=berkyear(9:12:end);
    annualAnomalyplot=annualAnomaly(9:12:end);
```

Plot Berkeley data

```
    % Create figure
    figure2 = figure;

    % Create axes
    axes2 = axes('Parent',figure2);
    hold(axes2,'on');

    % Create ylabel
    ylabel('Temperature Deviation from 1951-1980 Average (ºC)');

    % Create xlabel
    xlabel('Time (years)');

    % Create title
    title('Average Earth Temperature Per Year vs. Time (Source: BerkeleyEarth)');

    % Create scatter
```

```
scatter2 = scatter(berkyearplot(1:264),annualAnomalyplot(1:264));

% Get xdata from plot
xdata2 = get(scatter2, 'xdata');
% Get ydata from plot
ydata2 = get(scatter2, 'ydata');
% Make sure data are column vectors
xdata2 = xdata2(:);
ydata2 = ydata2(:);


% Remove NaN values and warn
nanMask2 = isnan(xdata2(:)) | isnan(ydata2(:));
if any(nanMask2)
    xdata2(nanMask2) = [];
    ydata2(nanMask2) = [];
end

% Find x values for plotting the fit based on xlim
axesLimits2 = xlim(axes2);
xplot2 = linspace(axesLimits2(1), axesLimits2(2));


fitResults2 = polyfit(xdata2, ydata2, 1);
% Evaluate polynomial
yplot2 = polyval(fitResults2, xplot2);
% Plot the fit
fitLine1 = plot(xplot2,yplot2,'DisplayName','   linear','Tag','linear',...
    'Parent',axes2,...
    'Color',[0.929 0.694 0.125]);

% Create legend
legend2 = legend(axes2,'show');
set(legend2,'Location','southeast','FontSize',9);
```



Analyze Berkeley data

```
berkyearplot=berkyear(9:12:end);
annualAnomalyplot=annualAnomaly(9:12:end);

fitResults2 = polyfit(berkyearplot, annualAnomalyplot, 1);
yplot2 = polyval(fitResults2, xplot2);
yresid2 = ydata2(1:100) - yplot2';
n_resid2=norm(yresid);
SSresid2 = sum(yresid.^2);
SStotal2 = (length(ydata2)-1)*ydata2;
rsq2 = 1 - SSresid2/SStotal2;
disp('RSq Value is:')
disp(rsq2(1))
```

```
RSq Value is:
     1
```

Analysis

```
% Both the NOAA and Berkeley datasets support the data discussed in
% question three. The NOAA plot indicates that the average temperature of
% the earth has increased in the last hundred years and will continue with
% a rate of about 0.070 °C per year into the 21st century. The Berkeley
```

```
% plot indicates that the average temperature of the earth has increased in
% the last hundred years as well and will continue with a rate of about
% 0.067 ºC per year into the 21st century. These two rates, as well as the
% given, are very compatible.
```

Published with MATLAB® R2014b