Exploratory Data Analysis on Temperatures of the Middle East and its Implications on Future Human Activity

Bryan Coronel
Washington University in St. Louis
EECE 202: Computational Modeling in EECE
Dr. Janie Brennan

May 11th 2020

Introduction

The continuous emissions of greenhouse gases has caused the earth to experience consistent increases in temperature. Greenhouse gases such as carbon dioxide are able to absorb infrared radiation reflected by the Earth's surface, which originated from the Sun. The gases in the atmosphere act as a shielding layer, keeping the planet's surface warmer than it otherwise would be. Over time, the layer has grown and accordingly, so has the warming *(1)*.

Global Warming has been studied extensively and over time, mathematical models have been developed to forecast temperature changes. One study explored techniques such as logistic, non-linear, and linear regression models *(2)*. This report will test models using linear regression among other numerical methods to predict and explore the ramifications of temperature increases. A caveat to this exploration is that a region-specific approach to organizing data may yield more accurate results *(3)*. Additionally, expectations must also be tempered with the fact that there are general uncertainties to modelling a property of the climate such as temperature *(3)*. The purpose of forecasting is important to regions most susceptible to global warming such as the Middle East, which already contain uninhabitable areas that are expanding as a result of global warming *(4)*. One study concluded that mortality risk increases yearly *(4)*. The report will determine places experiencing the high temperatures and *rates* of temperature increase.

Studies on heat wave mortalities provide a useful baseline of long-term deadly temperatures; a common one found in two studies was 35°C *(5,6)*. For the rest of the report, analysis will be performed on the countries approaching this value the quickest. Based on the forecasting model, the amount of time left to reach that temperature will be also determined. In short, the aim of this report is to highlight the regions of the world most susceptible to global warming, on the fastest track to deadly temperatures and when they will inevitably reach them.

Description of the Data

The dataset being analyzed contains information on the daily average temperature of various countries during a timespan ranging from 1995-2020 *(7)*. To account for countries with varying climates, the set contains data on several cities for countries such as the United States of America, Australia, and China, where the "state" or "city" columns are used. This dataset was sourced from Kaggle.com, where it was originally compiled and cleaned by the University of Dayton. It contains some data that is not usable due to insufficient data existing for a certain date or location. Rows containing temperature values of -99 are observations that should be dropped.

Data Analysis and Numerical Methods

The first step in exploratory data analysis is typically data cleaning because it can prevent unwanted errors. However, the dataset was already cleaned. Recall that temperature values of -99°F signify unavailable data, so they were removed. Additionally, unique elements in the dataset were extracted per variable which yielded that the dataset contained data for 125 countries and 26 years.

Numerical curve-fitting was employed using two techniques;. Linear Regression was used to extrapolate temperature values outside of the range of time in the data; an explanation of the model chosen will be explained. Numerical root-finding was then used to forecast the year countries would reach deadly temperatures based on a monthly average. In a separate function, a piecewise interpolation function was built on values of average monthly temperature and years to interpolate the derivatives within the data's range of time. A cubic hermite interpolating polynomial was chosen because it could uphold past scientific studies of the increasing rate in temperature change. Differentiation was then performed using the numerical gradient of the piecewise polynomial because it assumes evenly-spaced data points, which is what the command was built for. A calculation of finite differences should be sufficient in terms of the order of accuracy. Numerical one-dimensional optimization was performed using a MATLAB command called fminbnd, which is based on the Golden Section Search and parabolic interpolation methods. This command is only able to find minima, so initialized functions must be reflected across the x-axis to find maxima, using the '-' operator. Recall that the derivative of the previously created piecewise polynomial will be optimized in order to find the year in which the rate of temperature change was the highest. These values along with others will then be compared to the literature to see if they can be corroborated.

Results and Discussion

This report sought to determine the crucial year countries approached a deadly average temperature of 95°F (35°C). Linear regression was chosen to forecast, so several models will be compared to establish legitimacy of the model used moving forward. Saudi Arabia and Qatar's July data will be used as an example. Figure 1 shows a plot of the indexed data points and their respective fitted line using an exponential model

Exponential Model: $ln(y) = ln(a_0) + a_1 * x$

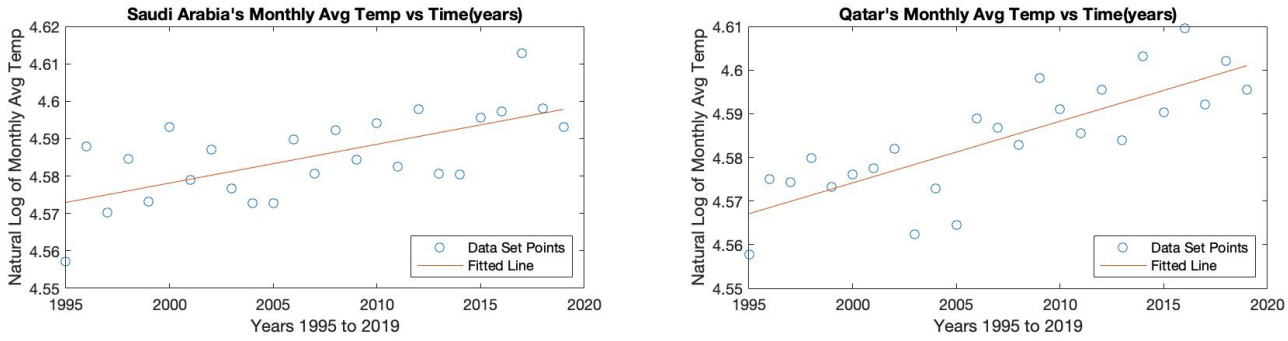


Fig. 1 Plot of the Natural Logarithm of Monthly Average Temperature versus Time in years ranging from 1995 to 2019 along with a fitted line based on an exponential model

The linear model appears to provide decent predictions. Figure 2 shows a plot of the indexed data points and their respective fitted line using a Saturation Kinetics model

Saturation Kinetics Model: $1/y = a_0 + a_1 * 1/x$

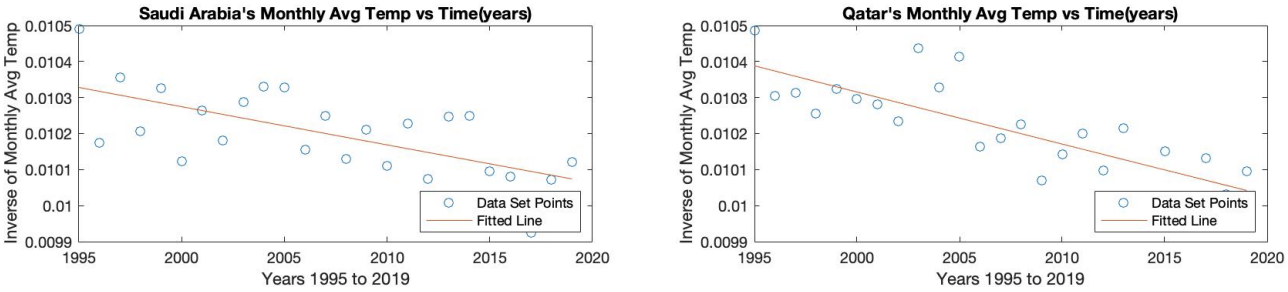


Fig. 2 Plot of the Inverse of Monthly Average Temperature versus Time in years ranging from 1995 to 2019 along with a fitted line based on a Saturation Kinetics model

The Saturation Kinetics model predicts decreasing temperatures over time, which is not corroborated by the literature, so this model can be immediately disregarded *(3)*. Figure 3 shows a plot of the indexed data points and their respective fitted line using a linear model

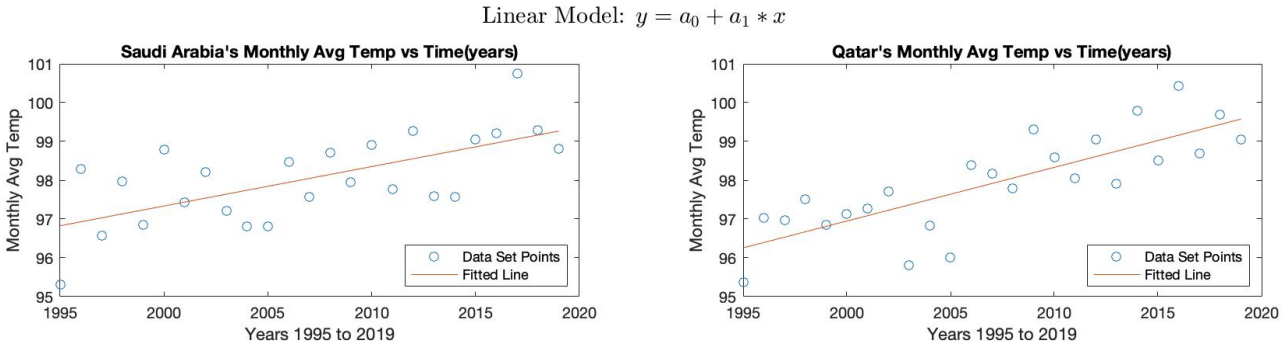


Fig. 3 Plot of the Monthly Average Temperature versus Time in years ranging from 1995 to 2019 along with a fitted line based on a linear model

The last model tested was the power law model and Figure 4 demonstrates the results

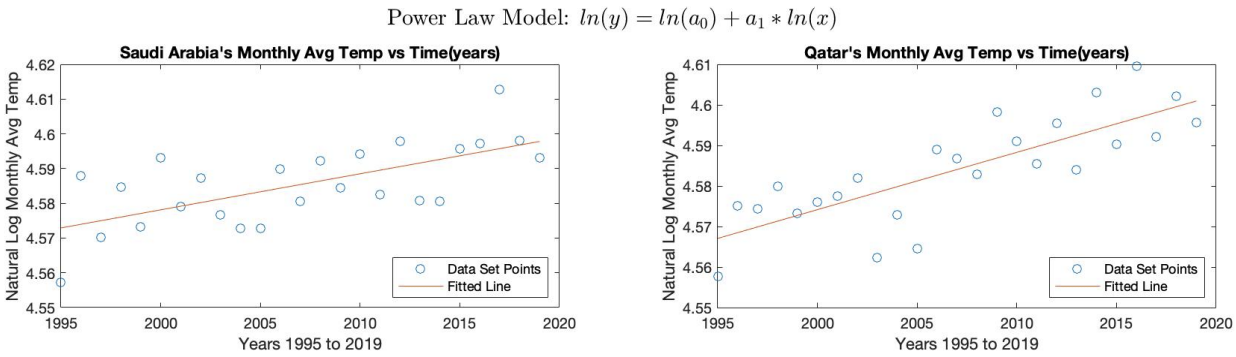


Fig. 4 Plot of the Monthly Average Temperature versus Time in years ranging from 1995 to 2019 along with a fitted line based on a power law model

A quantitative manner of assessing which model to choose is explored below in Tables 1, 2, and 3, where *S_r* is the sum of square residuals, *S_yx* is the standard error of estimate, *S_x* is the error from describing the data by its average, *R^2* is the coefficient of determination, and *R* is the correlation coefficient.

Table 1: Statistical Measurements of the Exponential Model

Country	S_r	S_yx	S_x	R^2	R
'Saudi Arabia"	0.0018624	0.0089987	0.0032607	0.42883	0.65485
'Qatar"	0.0014868	0.0080402	0.0040799	0.63558	0.79723

Table 2: Statistical Measurements of the Linear Model

Country	S_r	S_yx	S_x	R^2	R
Saudi Arabia"	17.886	0.88184	31.314	0.42884	0.65486
Qatar"	14.204	0.78585	39.074	0.63649	0.7978

Table 3: Statistical Measurements of the Power Law Model

Country	S_r	S_yx	S_x	R^2	R
'Saudi Arabia"	0.0018628	0.0089996	0.0032607	0.42871	0.65476
'Qatar"	0.0014871	0.0080409	0.0040799	0.63551	0.79719

Any of these models would be a decent choice given that the coefficient of determination values, *R^2*, are around 0.5, meaning a weak to moderate correlation. The Power Law Model was chosen and now a prediction can be made. Below in Table 4 are the years calculated from the root-finding method and data from September

Table 4: Predicted year for September to reach an average temperature of 95°F

Country	Fatal Year
"Saudi Arabia"	"2055.4307"
"Qatar"	"2039.7277"
"Kuwait"	"2029.9074"
"United Arab Emirates"	"2039.9097"
"Bahrain"	"2046.1762"
"Egypt"	"2155.8962"

The root-finding method calculated reasonable years in the distant future given that the literature suggests mortality risks will triple in the next 30 years *(4)*. Equally as important as predicting when countries will reach deadly temperatures is determining which countries are experiencing the highest levels of global warming. Below, Table 5 shows the highest rate at which middle eastern countries are warming between the period of 1995 and 2019.

Table 5: Highest rate of temperature increase between 1995 and 2019

Country	Year	Temp Rate (°F/yr)
"Saudi Arabia"	"2016"	"0.85322581"
"Qatar"	"2013"	"0.37580645"
"Kuwait"	"2017"	"0.61451613"
"United Arab Emirates"	"2010"	"1.3822581"
"Bahrain"	"2016"	"1.0258065"
"Egypt"	"2016"	"1.8403226"

The highest rates of temperature increases have occurred mostly in the past 5 years. This suggests that global warming is rapidly worsening, reaching rates predicted by time periods predicted earlier by the literature *(8)*. Furthermore, any potential outliers can largely be attributed to the fact that the models and techniques like linear regression are not as complex as the ones used by the current scientists researching this issue *(2)*. As a result, this report has been able to determine that countries like Kuwait and the United Arab Emirates are quickly running out of time to do something about harmful symptoms of global warming such as deadly heat waves.

Conclusion

Inevitably, all the Middle Eastern countries tested in this report will become uninhabitable. It can be seen that the United Arab Emirates and Kuwait have worrying rates of temperature increase and dates that are not too far in the future. However, other aspects of the climate will also be exacerbated such as patterns of precipitation, worsening natural disasters, etc *(8)*. In summary, this report only tackles only one symptom of climate change, but there are many more to worry about, especially in the Middle East.

References

(1) Houghton, J. Reports on Progress in Physics Global Warming; 2005.

(2) Mohseni, O.; Erickson, T. R.; Stefan, H. G. Sensitivity of Stream Temperatures in the United States to Air Temperatures Projected under a Global Warming Scenario. Water Resour. Res. 1999, 35 (12), 3723–3733.

(3) Karmalkar, A. V.; Bradley, R. S. Consequences of Global Warming of 1.5 °C and 2 °C for Regional Temperature and Precipitation Changes in the Contiguous United States. PLoS One 2017, 12 (1), e0168697.

(4) Ahmadalipour, A.; Moradkhani, H. Escalating Heat-Stress Mortality Risk Due to Global Warming in the Middle East and North Africa (MENA). Environ. Int. 2018, 117, 215–225.

(5) Tan, J.; Zheng, Y.; Song, G.; Kalkstein, L. S.; Kalkstein, A. J.; Tang, X. Heat Wave Impacts on Mortality in Shanghai, 1998 and 2003. Int. J. Biometeorol. 2007, 51 (3), 193–200.

(6) Bi, P.; Williams, S.; Loughnan, M.; Lloyd, G.; Hansen, A.; Kjellstrom, T.; Dear, K.; Saniotis, A. The Effects of Extreme Heat on Human Mortality and Morbidity in Australia: Implications for Public Health. Asia. Pac. J. Public Health 2011, 23 (2 Suppl), 27S – 36.

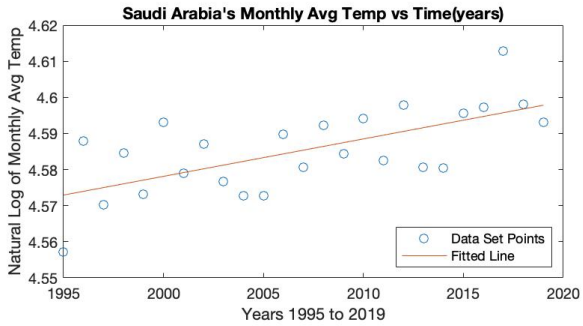
(7) SRK. Daily Temperature of Major Cities.

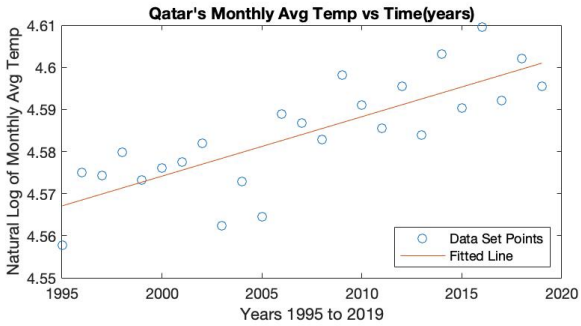
(8) Evans, J. P. 21st Century Climate Change in the Middle East. Clim. Change 2009, 92 (3–4), 417–432.

Appendix A: Tables and Figures

A1

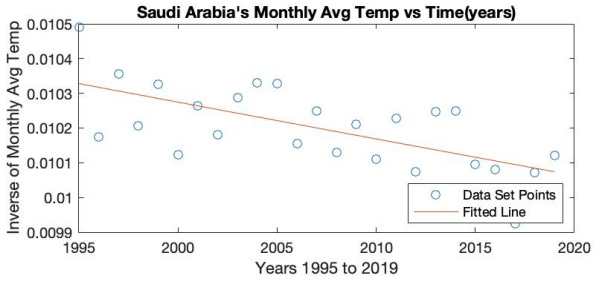
Exponential Model: $ln(y) = ln(a_0) + a_1 * x$

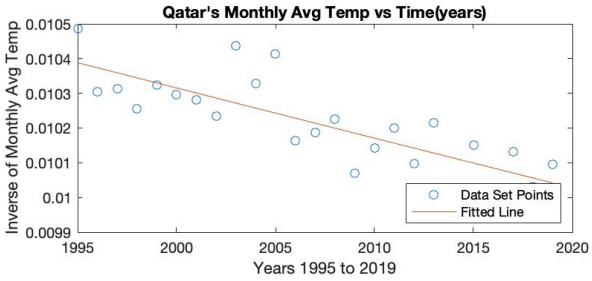




A2

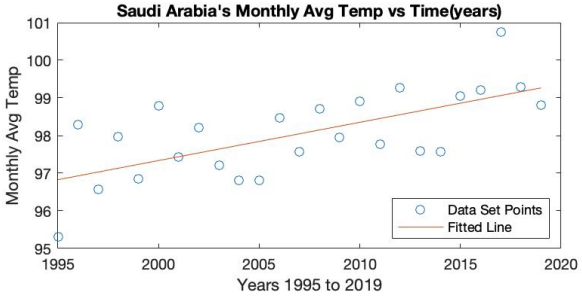
Saturation Kinetics Model: $1/y = a_0 + a_1 * 1/x$

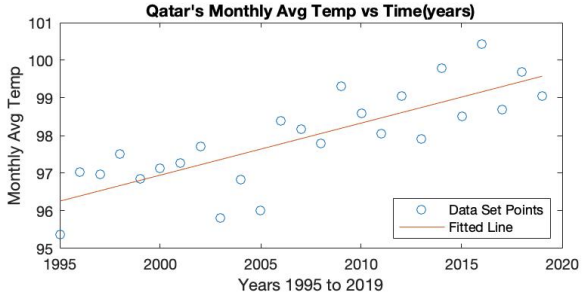




A3

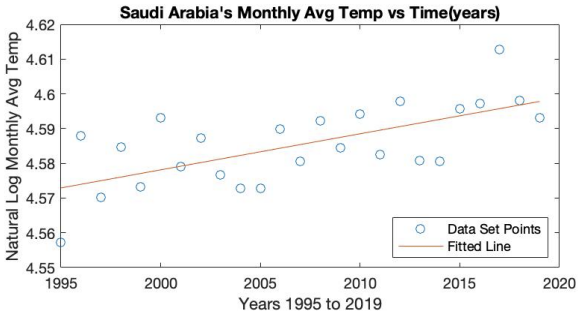
Linear Model: $y = a_0 + a_1 * x$

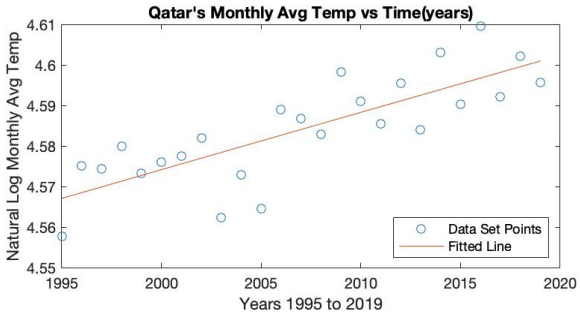




A4

Power Law Model: $ln(y) = ln(a_0) + a_1 * ln(x)$





A5

Country	S_r	S_yx	S_x	R^2	R
'Saudi Arabia"	0.0018624	0.0089987	0.0032607	0.42883	0.65485
'Qatar"	0.0014868	0.0080402	0.0040799	0.63558	0.79723

A6

Country	S_r	S_yx	S_x	R^2	R
Saudi Arabia"	17.886	0.88184	31.314	0.42884	0.65486
Qatar"	14.204	0.78585	39.074	0.63649	0.7978

A7

Country	S_r	S_yx	S_x	R^2	R
'Saudi Arabia"	0.0018628	0.0089996	0.0032607	0.42871	0.65476
'Qatar"	0.0014871	0.0080409	0.0040799	0.63551	0.79719

A8

Country	Fatal Year
"Saudi Arabia"	"2055.4307"
"Qatar"	"2039.7277"
"Kuwait"	"2029.9074"
"United Arab Emirates"	"2039.9097"
"Bahrain"	"2046.1762"
"Egypt"	"2155.8962"

A9

Country	Year	Temp Rate (°F/yr)
"Saudi Arabia"	"2016"	"0.85322581"
"Qatar"	"2013"	"0.37580645"
"Kuwait"	"2017"	"0.61451613"
"United Arab Emirates"	"2010"	"1.3822581"
"Bahrain"	"2016"	"1.0258065"
"Egypt"	"2016"	"1.8403226"

Appendix B: MATLAB Code

Table of Contents


```
clear; close all; clc; format compact;
% NOTE: the file is separated by sections but can be run as a whole. A
 lot
% of output will be spit out in the command window. All of it is
 relevant.
% NOTE: please resize graphs as you see fit, I cannot account for how
 plots
% look on different computers.
% NOTE: I hope you enjoy looking at my work. I enjoyed analyzing
% this data so much that I'm publishing this project to my github!
% importing csv file as a table, so that it is easier to splice and
 index
global T month country_array;
T = readtable('city_temperature.csv');
```

exploring

```
clearvars -except T; close all; clc; format compact;
% Check out the function file to see my comments and code!
T = exploring(T);
```

regression and root-finding

```
clearvars -except T & country_array & month; close all; clc;
 format compact;
% I initialized some values for the parameters of my functions for
 you, but
% feel free to changed them to see that I didn't "hard-code" anything
month = 9; forecast_year = 2019;
country_array = ["Saudi Arabia", "Qatar", "Kuwait", "United Arab
 Emirates", "Bahrain", "Egypt"];
stats = linreg_rootfinding(T, country_array, month,forecast_year)
```

splines, differentiation and optimization

```
month = 7;
country_array = ["Saudi Arabia", "Qatar", "Kuwait", "United Arab
 Emirates","Bahrain", "Egypt"];
rate = spline_diff_opt(T,country_array,month)
```

Published with MATLAB® R2020a

```
% THIS IS THE START OF A NEW FUNCTION
% exploring: function T = exploring(T)
%
% This function outputs unique values of your table to see what the
% variables contained are
% output:
%   T = a modified table with dropped rows of data that is not
 sufficient
% input:
%   T = the original table that was imported initially
function T = exploring(T)
% exploring features and their general statistics
data_summary = summary(T);
% how many observations do we have originally?
observations = height(T)
% From data set description, temperature values of -99 signify no data
 was availble for that date
% I will drop these rows via logical indexing so that they don't
 interfere with plots and MATLAB commands
% condition I'm looking for
toDelete = T.AvgTemperature ==-99 | T.Year == 200 | T.Year == 201;
% the deletion
T(toDelete,:) = [];
% checking to see how many observations I have now
observations_after_drop = height(T)
% out of curiosity, how many observations did we lose? presented as a
 percentage
lostrows = abs(observations_after_drop-observations)/observations*100
% here I'm exploring the unique regions in the data set
regions = unique(T(:,1))
% here I'm exploring the years for which the data is recorded
years = unique(T(:,7))
% here I'm trying to see which cities are in the set
cities = unique(T(:,4));
% here I'm trying to see which countries are in the set
countries = unique(T(:,2))
end
```

Published with MATLAB® R2020a

```
% THIS IS THE START OF A NEW FUNCTION
% linereg_footfinding: function stats =
 linreg_rootfinding(T,country_array,month,forecast_year)
%
% This function implements several numerical methods
% Employs linear regression using the power law model to forecast
 future
% temperatures. Uses root-finding to make a prediction of when a
 certain
% temperature will be reached
%
% output:
%   stats = table that contains statistical measurements of the model
%   used to see how well it fit the data
% input:
%   T = table of data that must be imported first
%   country_array = is a string array that contains country names,
 make
%   sure they're actually in your table though
%   month = choose which month will get indexed in the table
%   forecast_year = choose until what year you want plot your fitted
%   regression line to forecast some temperatures

function stats =
 linreg_rootfinding(T,country_array,month,forecast_year)
% initializing a row vector containing years for which data is
 available
% Note: 2020 data was not sufficient for some Middle Eastern countries
year = 1995:2019;
% although not needed typically, these arrays were created because in
 some
% cases MATLAB would not work without them :(
stats_array = ones(length(country_array),5);
death_year_array = ones(1,length(country_array));
% a loop that will index into every country's in the string array
for i=1:length(country_array)
    % Using logical indexing and string comparison for the string
 array
    % that must be passed into the function, this code will return the
    % indices of the matching criteria of country and month, which can
 be
    % specified outside the function
    index = find(strcmp(country_array(i), T.Country) &  T.Month ==
 month);
    % Here is where I store the indexed data
    indexed_data = T(index,:);
    % Another loop where I extract the indices of the values in the
 average temperature
    % column. These indices are then used to generate another table to
 then convert it to an array,
    % where I take the mean of the array. This is the average
 temperature
```

```
    % for the month.
    for j=1:length(year)
        year_avg(j) =
mean(table2array(indexed_data(find(indexed_data.Year==(1994+j)),8)));
    end
    % Used this command so that you get 6 figures
    figure()
    % Here I created subplots, so that you can see all the data points
    % extracted with their respective fitted lines on top. Commented
out
    % because of the way it outputs
    %subplot  (3,2,i)
    % Here I make use of the rmoutliers() command in case some years
are
    % anomalies in terms of the overall climate
    [year_avg, remove_year] = rmoutliers(year_avg,'mean');
    % Clearing the value of the year where the anomaly appeared, so
that
    % the dimensions of the matrices agree when performing regression
    year(remove_year==1)=[];
    % plot log(y) vs x
    plot(year,log(year_avg), 'o');
    % appropriate plot components
    title(country_array(i) + '''s Monthly Avg Temp vs Time(years)')
    xlabel('Years 1995 to 2019'), ylabel('Natural Log Monthly Avg
Temp')
    hold on
    % LINEAR REGRESSION
    % power law model ln(y) = ln(a_0) + a_1*ln(x)
    % Here I initialized my matrices to least squares perform
regression
    Z = [ones(length(year),1) log(year)']; y = (log(year_avg))'; coeff
= Z\y;
    % I initiliazed some year values to make the fitted line look
smoth
    xvals =  linspace(1995, forecast_year, 1000);
    % plotting my fitted line with appropriate components
    plot(xvals,coeff(1)+ coeff(2).*log(xvals));
    legend('Data Set Points', 'Fitted Line','Location', 'southeast')
    hold off
    format short
    % Here I make use of a loop to create an array to store some
measure of
    % how good my fitted line is
    for k=1:5
        if (k==1)
            % sum of square residuals
            S_r = sum((y-Z*coeff).^2); stats_array(i,k) = S_r;
        end
        if (k==2)
            % standard error of estimate
            S_yx = sqrt( S_r /(length(year)-length(coeff)));
stats_array(i,k) = S_yx;
        end
```

```
            if (k==3)
                % total sum of squares
                S_t = sum((y - mean(y)).^2); stats_array(i,k) = S_t;
            end
            if (k==4)
                %coefficient of determination
                r_squared = 1 - (S_r/S_t); stats_array(i,k) = r_squared;
            end
            if (k==5)
                % correlation coefficient
                r = sqrt(r_squared); stats_array(i,k) = r;
            end
        end
        % To make for cleaner output, I converted it to a table
        stats_table = array2table(stats_array);
        % concatenated a column containing the country names
        stats = [array2table(country_array'), stats_table];
        % adding variable names to table
        stats.Properties.VariableNames =
 {'Country' 'S_r' 'S_yx' 'S_' 'R^2' 'R'};
        % ROOT-FINDING
        % initializing the temperature at which living in for multiple
        % consecutive days can be deadly!
        death_temp = 98;
        % creating an array where I store the root of setting my fitted
        % regression model equal to the deadly temperature. Recall my
 dependent
        % variable must have exp() used on it, given I linearized my model
        death_year_array(i) = exp(roots([coeff(2), coeff(1)-
log(death_temp)]));
        death_year = array2table([country_array' death_year_array']);
        % adding variable names to table
        death_year.Properties.VariableNames = {'Country' 'Fatal Year'};
    end
    % If plotting using subplot(), uncomment
    % sgtitle('Power Law Model: $$ ln(y) = ln(a_{0}) + a_{1}\ast ln(x)$
    $','interpreter', 'latex')
    % not supressing my table
    death_year
    end

    % Note: Here is the code I constructed to test other models out. They
     will
    % not be a part of my output because they are not the final model I
     ended
    % up choosing!
    % linear model
    %Z = [ones(length(year),1) year']; y = year_avg'; coeff = Z\y;
    %xvals = linspace(1995,forecast_year , 1000);
    %plot (xvals, coeff(1)+coeff(2).*xvals)

    % saturation kinetics
    %Z = [ones(length(year),1) (1./year)']; y = (1./year_avg)'; coeff = Z
    \y;
```

```
%xvals = linspace(1995, forecast_year, 1000); yvals = coeff(1) +
 coeff(2).* (1./xvals);
%plot(xvals, yvals)

% exponential model
%Z = [ones(length(year),1) year']; y = (log(year_avg))'; coeff = Z\y;
%xvals = linspace(1995, forecast_year, 1000); yvals = coeff(1) +
 coeff(2).*xvals;
%plot(xvals, yvals);
```

Published with MATLAB® R2020a

```
% THIS IS THE START OF A NEW FUNCTION
% spline_diff_opt: function rate =
 spline_diff_opt(T,country_array,month)
%
% This function employs the use of splines to differentiate and
 optimize
% them to explore rates of temperature change per year
% output:
%   rate = a table of the rates of temperature change per country
% input:
%   T = table of data that must be imported first
%   country_array = is a string array that contains country names,
 make
%   sure they're actually in your table though
%   month = choose which month will get indexed in the table
function rate = spline_diff_opt(T,country_array,month)
% initializing a row vector containing years for which data is
 available
% Note: 2020 data was not sufficient for some Middle Eastern countries
year = 1995:2019;
% although not needed typically, these arrays were created because in
 some
% cases MATLAB would not work without them :(
rate_array = ones(length(country_array),2);
% a loop that will index into every country's in the string array
for i=1:length(country_array)
    % Using logical indexing and string comparison for the string
 array
    % that must be passed into the function, this code will return the
    % indices of the matching criteria of country and month, which can
 be
    % specified outside the function
    index = find(strcmp(country_array(i), T.Country) &  T.Month ==
 month);
    % Here is where I store the indexed data
    indexed_data = T(index,:);
    % Another loop where I extract the indices of the values in the
 average temperature
    % column. These indices are then used to generate another table to
 then convert it to an array,
    % where I take the mean of the array. This is the average
 temperature
    % for the month.
    for j=1:length(year)
        year_avg(j) =
 mean(table2array(indexed_data(find(indexed_data.Year==(1994+j)),8)));
    end
    % DIFFERENTIATION
    % Here I make use of the gradient() command since I have evenly-
spaced
    % data!
    myderivative = gradient(year_avg, 1);
```

```
    % Used this command so that you get 6 figures
    figure()
    %subplot  (3,2,i)
    % plotting the derivative to see how rate of temperature change
 changes
    % over time
    plot(year, myderivative,'o')
    hold on
    title(country_array(i) + '''s Rate of Temp of Change vs
 Time(years)')
    xlabel('Years 1995 to 2019'), ylabel('Change in Temp per Year')
    % creating a piecewise interpolating polynomial from the data
 using PCHIP
    % to model physical phenomena correctly
    f = @(xx) interp1(year,myderivative, xx,'PCHIP');
    % plotting piecewise polynomial
    fplot(f,[1995 2019]);
    % OPTIMIZATION
    % using fminbnd() on the negative version of my piecewise
 polynomial to
    % find where the rate of temperature change is highest
    [xmax, negFmax] = fminbnd(@(xx) -f(xx), 2005,2019);
    % storing these years in an array in order to create a neat table!
    rate_array(i,1) = xmax; rate_array(i,2) = -negFmax;
    % coverting to table and concatenating a column of the countries
    rate = array2table([country_array' rate_array]);
    % adding variables names to the table
    rate.Properties.VariableNames = {'Country' 'Year' 'Temp Rate (°F/
yr)'};
end
rate
end
```

Published with MATLAB® R2020a