Homework 4

Terror Bird Weight based on present day birds:

% enter the femur circumfrence(cm) values in a row matrix

L = [0.7079, 0.7943, 1.000, 1.122, 1.6982, 1.2023, 1.9953, 2.2387, 2.5119, 3.1623, 3.5481, 4.4668, 5.8884, 6.7608, 15.136, 15.85];

%enter the body weight(kg) values in a row matrix

W = [0.0912, 0.0832, 0.1413, 0.1479, 0.2455, 0.2818, 0.7943, 2.5119, 0.8913, 1.9953, 4.2658, 6.3096, 11.2202, 19.95, 141.25, 158.4893];

%establish the function that will find the proportionality constant

phi = @(k) (sum(((W-k\*L.^3)).^2));

% find the value that minimizes k

[k,fval]=fminsearch(phi,1)

% value is k = 0.0403, fval = 89.9093

% define the new function NW = calculated weight

NW = k\*L.^3;

%plot the new function with the old data points to visualize error

h = figure;

plot(L, W,'+', L, NW, '--')

saveas(h,sprintf('FIG%d.png',1)); % will create FIG1, FIG2,...

%generalize the function to predict new weights, PW

l = [1:.1:max(L)];

PW = k\*l.^3;

%plot the generalized function with the modeled function, and original data

%points

g = figure;

plot(l,PW, ':')

saveas(g,sprintf('FIG%d.png',2)); % will create FIG1, FIG2,...

%find the value of the Titanis Walleri

k\*(21)^3

Terror Bird Weight based on dinosaur records

% enter the femur circumfrence(cm) values in a row matrix

L = [10.3, 13.6, 20.1, 26.7, 34.8, 40, 50.4, 51.2, 53.4];

%enter the body weight(kg) values in a row matrix

W = [55, 115, 311, 640, 1230, 1818, 3300, 3500, 4000];

%establish the function that will find the proportionality constant

phi = @(k) (sum(((W-k\*L.^3)).^2));

% find the value that minimizes k

[k,fval]=fminsearch(phi,1)

% k = 0.0264 fval = 7.0119X10^4

% define the new function NW = calculated weight

NW = k\*L.^3;

%plot the new function with the old data points to visualize error

h = figure;

plot(L, W,'+', L, NW, '--')

saveas(h,sprintf('FIG%d.png',3)); % will create FIG1, FIG2,...

%generalize the function to predict new weights, PW

l = [1:.1:max(L)];

PW = k\*l.^3;

%plot the generalized function with the modeled function, and original data

%points

g = figure;

plot(l,PW, ':')

saveas(g,sprintf('FIG%d.png',4)); % will create FIG1, FIG2,...

%find the value of the Titanis Walleri

k\*(21)^3

First Order Difference Equation/Sheep Population

3)

function [t, Y] = differenceEquation(f, T, y0)

%establishing the transformation matrix of birth, B, and death rates, D

B = [0, 0.045, 0.391, 0.472, 0.484, 0.546, 0.543, 0.502, 0.468, 0.459, 0.433, 0.421];

D = [0.845, 0.975, 0.965, 0.95, 0.926, 0.895, 0.85, 0.786, 0.691, 0.561, 0.37, 0];

D\_1 = [D(1), 0,0,0,0,0,0,0,0,0,0,0];

D\_2 = [0,D(2),0,0,0,0,0,0,0,0,0,0];

D\_3 = [0,0,D(3),0,0,0,0,0,0,0,0,0];

D\_4 = [0,0,0,D(4),0,0,0,0,0,0,0,0];

D\_5 = [0,0,0,0,D(5),0,0,0,0,0,0,0];

D\_6 = [0,0,0,0,0,D(6),0,0,0,0,0,0];

D\_7 = [0,0,0,0,0,0,D(7),0,0,0,0,0];

D\_8 = [0,0,0,0,0,0,0,D(8),0,0,0,0];

D\_9 = [0,0,0,0,0,0,0,0,D(9),0,0,0];

D\_10 = [0,0,0,0,0,0,0,0,0,D(10),0,0];

D\_11 = [0,0,0,0,0,0,0,0,0,0,D(11),0];

%concatinate the matricies into one large transformation matrix, A

A = [B; D\_1; D\_2; D\_3; D\_4; D\_5; D\_6; D\_7; D\_8; D\_9; D\_10; D\_11];

f = @(t,Y) (A\*Y);

%y0 is the initial conditions

%T is a two element matrix that determines the starting and end point of

%the difference equation

%f is the function handle

%define the value space for Y

Y = zeros(length(y0),T(2));

t = [T(1):T(2)-1];

%define the first value of Y to be the initial conditions

Y(:,1) = y0;

for j = (T(1)+2):T(2)

Y(:,j) = f(j-1,Y(:,j-1));

end

4)

using the function that was created in question 3

%define the function handle that will determine the difference equation

f = @(t,Y) (A\*Y);

%values of the start and end value for time

T = [0, 100];

%values of the initial conditions

y0 = [1;2;3;4;5;6;7;8;9;10;11;12];

%this spits out a vector with the time intervals and the values of the

%state variables

[t, Y] = differenceEquation(f, T, y0);

plot(t, Y(1,:),t,Y(2,:),t,Y(3,:),t,Y(4,:),t,Y(5,:),t,Y(6,:),t,Y(7,:),t,Y(8,:),t,Y(9,:),t,Y(10,:),t,Y(11,:),t,Y(12,:))

Total = zeros(12, T(2));

for u = 1:T(2)

Total(1,u) = Y(1,u)/(sum(Y(:,u)));

Total(2,u) = Y(2,u)/(sum(Y(:,u)));

Total(3,u) = Y(3,u)/(sum(Y(:,u)));

Total(4,u) = Y(4,u)/(sum(Y(:,u)));

Total(5,u) = Y(5,u)/(sum(Y(:,u)));

Total(6,u) = Y(6,u)/(sum(Y(:,u)));

Total(7,u) = Y(7,u)/(sum(Y(:,u)));

Total(8,u) = Y(8,u)/(sum(Y(:,u)));

Total(9,u) = Y(9,u)/(sum(Y(:,u)));

Total(10,u) = Y(10,u)/(sum(Y(:,u)));

Total(11,u) = Y(11,u)/(sum(Y(:,u)));

Total(12,u) = Y(12,u)/(sum(Y(:,u)));

end

plot(t, Total(1,:),t,Total(2,:),t,Total(3,:),t,Total(4,:),t,Total(5,:),t,Total(6,:),t,Total(7,:),t,Total(8,:),t,Total(9,:),t,Total(10,:),t,Total(11,:),t,Total(12,:))

Final = (Total(:,T(2)))';

bar(Final)

%to find the growth rate, you take the inital population and set it equal

%to the initial condition, and then minimize the function that to find the

%value of b

a = TotalPopulation(1);

phi = @(b) (sum((TotalPopulation-a\*b.^t).^2));

[b, fval] = fminsearch(phi,1)

%b = 1.1709