Problem 1

Authors: Tony Okeke
Nick Corrado
Ben Jennings
Gabriella Grym
Date: 09.25.2020
Team: A_03

Calculate Temperature at Four Locations

Script Description

- This script determines the temperatures at different locations based on the model :
  y(t) = R.sin(2*pi/24*(t-t0)) + B
- It returns plots of temperature against time at these locations.

```
% Define time variable
   t_data = (0:24)';

% Define cnsts table containing constants at different locations
   location = {'Philadelphia, PA','Panama City, FL','Brewster, MA',...
             'Enugu, Nigeria'}';
  eq_cnst = [25 9 55 ; 10 9 72 ; 15 9 59 ; 13 9 73];
  % eq_cnst(:,1) -> R (oF)
  % eq_cnst(:,2) -> t0 (hrs)
  % eq_cnst(:,3) -> B (oF)
  cnsts = table(location,eq_cnst);
```

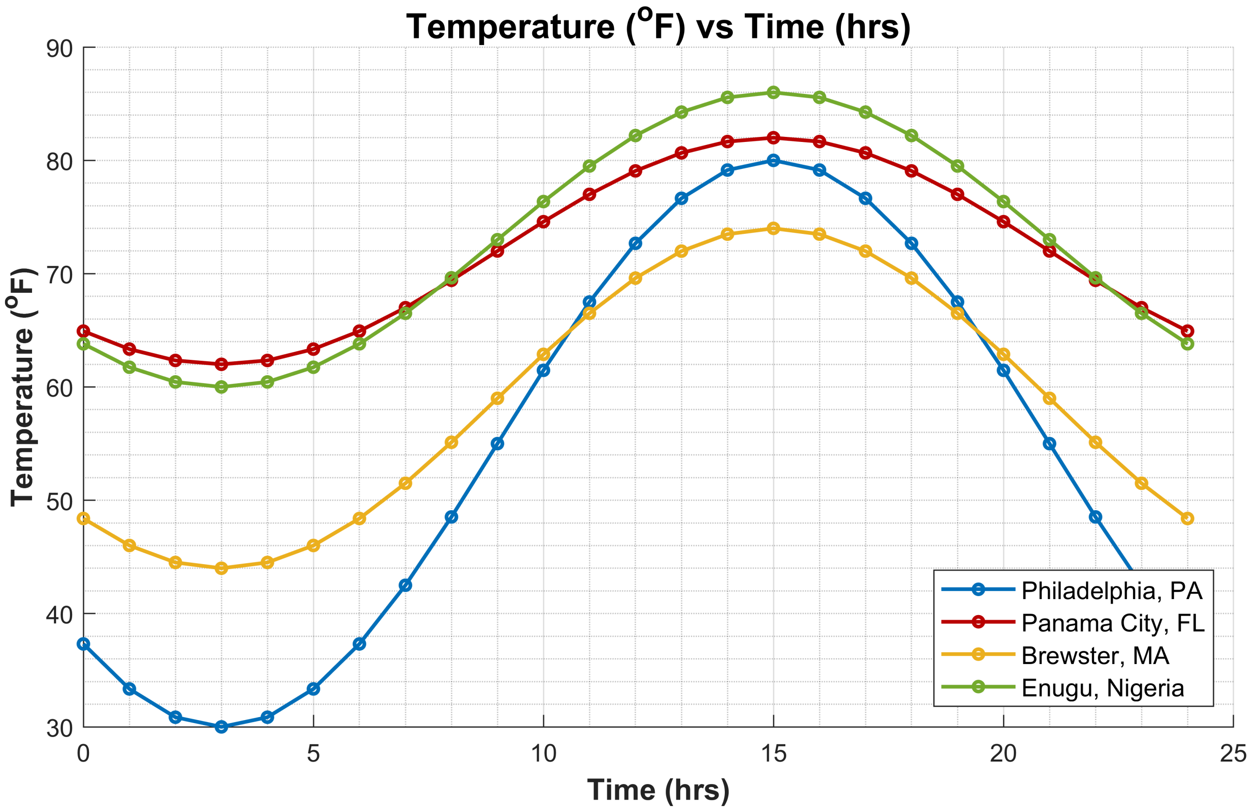
cnsts = 4×2 table

	location	eq_cnst		
1	'Philadelphia...	25	9	55
2	'Panama City, FL'	10	9	72
3	'Brewster, MA'	15	9	59
4	'Enugu, Nigeria'	13	9	73

Use ca1_fun1 to determine temperatures and plot Temperature vs Time graphs for each location.

```
% Configure Plot Area
  figure
  ax = axes;
 % Color order for the various graphs
 ax.ColorOrder = [0 0.447058823529412 0.741176470588235;0.729411764705882 0 0;...
          0.929411764705882 0.694117647058824 0.125490196078431;...
          0.466666666666667 0.674509803921569 0.188235294117647];
 % Adjust plot settings
 ax.XGrid = 'on';
 ax.XMinorGrid = 'on';
 ax.YMinorGrid = 'on';
 ax.Parent.Position = [900 450 800 450]; % window size
 hold on
  title('Temperature (^oF) vs Time (hrs)','FontWeight','bold','FontSize',14)
  xlabel('Time (hrs)','FontWeight','bold','FontSize',12)
 ylabel('Temperature (^oF)','FontWeight','bold','FontSize',12)

% Generate Graphs
  for i = 1:numel(location)
```

```
    % Calculate yt_data
      yt_data = ca1_fun1(cnsts.eq_cnst(i,:),t_data);
    % Plot Results
      plot(t_data,yt_data,'-o','LineWidth',1.5,'MarkerSize',4)
  end
   legend(location,'Location','southeast','FontSize',10)
  hold off
```



Determining Constants for Figure 1 (In Problem Statement)

Define Constants and t_data.

From Figure 1 (Problem Statement)

- Max Temp - Min Temp = 40 oF
- Range, R = 40/2 = ± 20 oF
- Bias, B = Max Temp - R = 60 oF

To determine the appropriate t0 value, we test all possible values and select the one that produces a graph matching figure 1

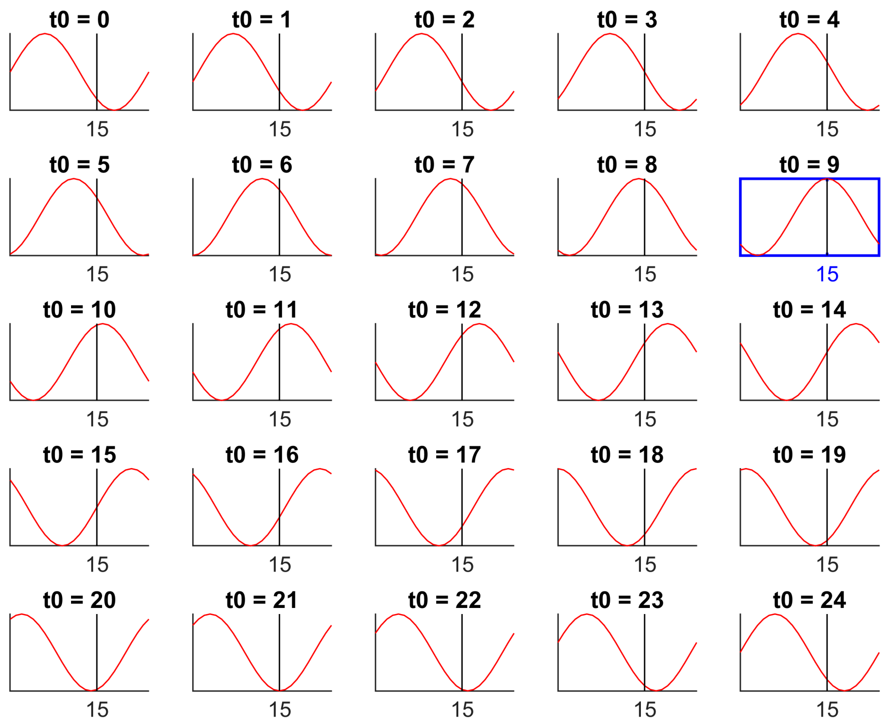
```
  R = 20;
  t0 = (0:24)'; % range of values for testing
  B = 60;

  t_data = (0:24)';
```

Calculate yt_data, and plot graphs for each potential t0 value in a 5 x 5 subplot grid.

```
figure
clf
for i = 1:numel(t0)
 % Calculate temperature
 yt_data = ca1_fun1([R t0(i) B],t_data);

 % Plot Graphs
 subplot(5,5,i)
 hold on
 title(sprintf('t0 = %i',t0(i)))
 plot(t_data,yt_data,'-r')
 plot([15 15],[40 80],'-k') % Identifies peak from problem statement
 xticks(15)
 yticks([])
 hold off
end
```

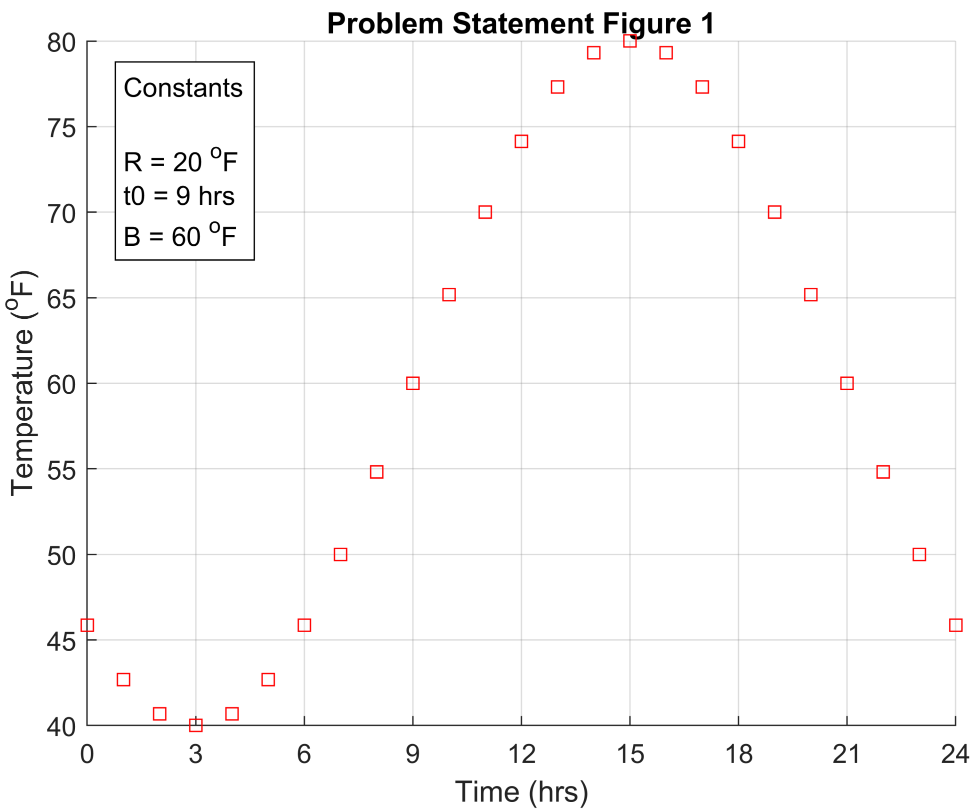
The graph t0 = 9 best matches figure 1 as it has the same peak at 15 hrs, as well as matching the shape of figure 1.

```
%   Emphasize the plot t0 = 9
   ax = subplot(5,5,10);
  box on
  ax.LineWidth = 1;
  ax.XColor = 'blue';
  ax.YColor = 'blue';
```



Generate Figure 1 From The Problem Statement

```
%   Plot Graph
    figure
  ax = axes;
  hold on
  plot(t_data,ca1_fun1([R t0(10) B],t_data),'sr');
  ax.XGrid = 'on';
  ax.YGrid = 'on';
  title('Problem Statement Figure 1')
  xlabel('Time (hrs)')
  ylabel('Temperature (^oF)')
  xlim([0 24])
  xticks([0 3 6 9 12 15 18 21 24])
%   Include a text box containing the determined constants
    txt = sprintf('Constants\n\nR = %i ^oF \nt0 = %i hrs \nB = %i ^oF',...
               R,t0(10),B);
    a = text(1,73,txt);
  a.EdgeColor = 'k';
  a.BackgroundColor = 'white';
  hold off
```



Problem 2

Calculate Heart-rate Decays for Three Subjects

Script Description

- This function models how the heart rate slows as it returns to 'normal' after exercise based on the model: **y(t) = D*exp(-t/tau) + B**
- It plots the return to baseline for three subjects (fit,average,unfit).

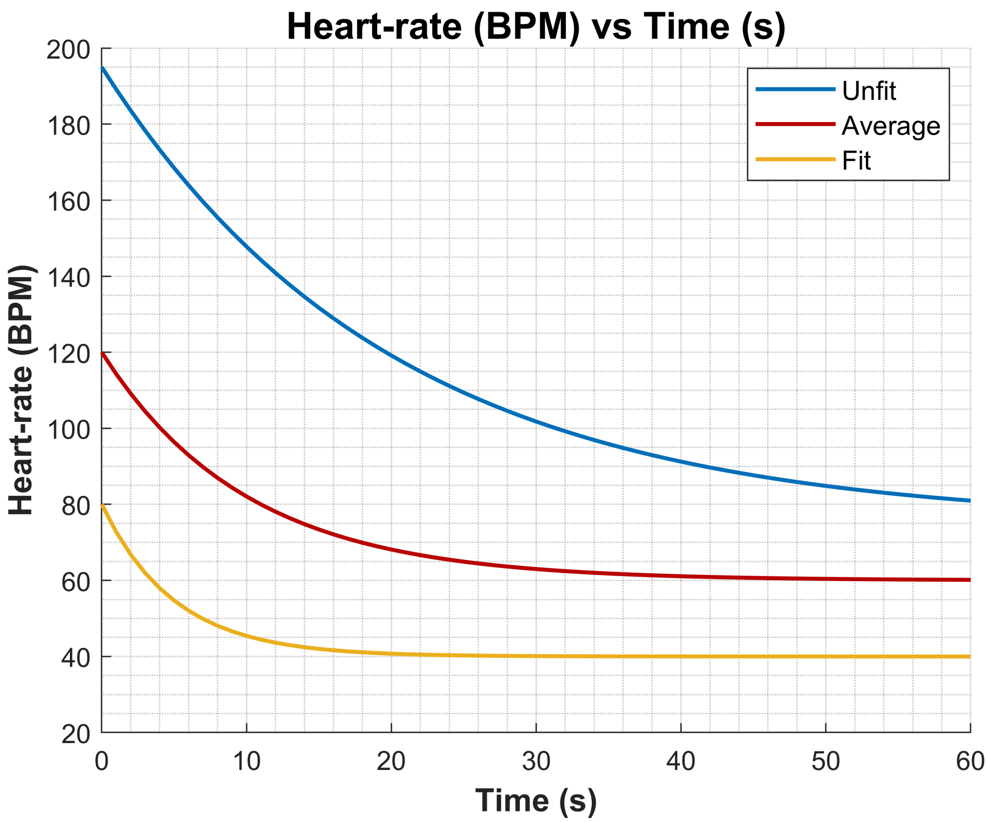
```
% Define time variable
   t_data = (0:60)';

% Define cnsts table containing constants for the different subjects
   fitness = {'Unfit','Average','Fit'}';
  eq_cnst = [120 20 75;60 10 60;40 5 40];
  % eq_cnst(:,1) -> D (BPM)
  % eq_cnst(:,2) -> tau (sec)
  % eq_cnst(:,3) -> B (BPM)
  cnsts = table(fitness,eq_cnst);
```

cnsts = 3×2 table

	fitness	eq_cnst		
1	'Unfit'	120	20	75
2	'Average'	60	10	60
3	'Fit'	40	5	40

Use ca1_fun2 to determine heart-rates and plot Heart-rate vs Time graphs for each subject.

```
% Configure Plot
  figure
  ax = axes;
 % Color order for the various graphs
 ax.ColorOrder = [0 0.447058823529412 0.741176470588235;0.729411764705882 0 0;...
          0.929411764705882 0.694117647058824 0.125490196078431];
 % Adjust plot settings
 ax.XGrid = 'on';
 ax.XMinorGrid = 'on';
 ax.YMinorGrid = 'on';
 hold on
  title('Heart-rate (BPM) vs Time (s)','FontWeight','bold','FontSize',14)
  xlabel('Time (s)','FontWeight','bold','FontSize',12)
 ylabel('Heart-rate (BPM)','FontWeight','bold','FontSize',12)
 yticks([20 40 60 80 100 120 140 160 180 200 220])

% Generate Graphs
  for i = 1:numel(fitness)
   % Calculate yt_data
     yt_data = ca1_fun2(cnsts.eq_cnst(i,:),t_data);
   % Plot Results
  plot(t_data,yt_data,'-','LineWidth',1.5)
 end
 ylim([20 200])
  legend(fitness,'Location','northeast','FontSize',10)
 hold off
```



Determining Constants for Figure 2 (In Problem Statement)

Define Constants and t_data.

From Figure 2 (Problem Statement)

- Increase in HR, D = Max HR - Min HR = 120 BPM
- Base-line HR, B = Min HR = 60 BPM

To determine the appropriate tau value, we test four different tau values (based on the values for fit,unfit,and average subjects). The tau value that produces a graph matching figure 2 is then selected.

```
D = 120;
tau = (5:5:20)'; % t0 contains a range of values for testing
B = 60;

t_data = (0:60)';
```

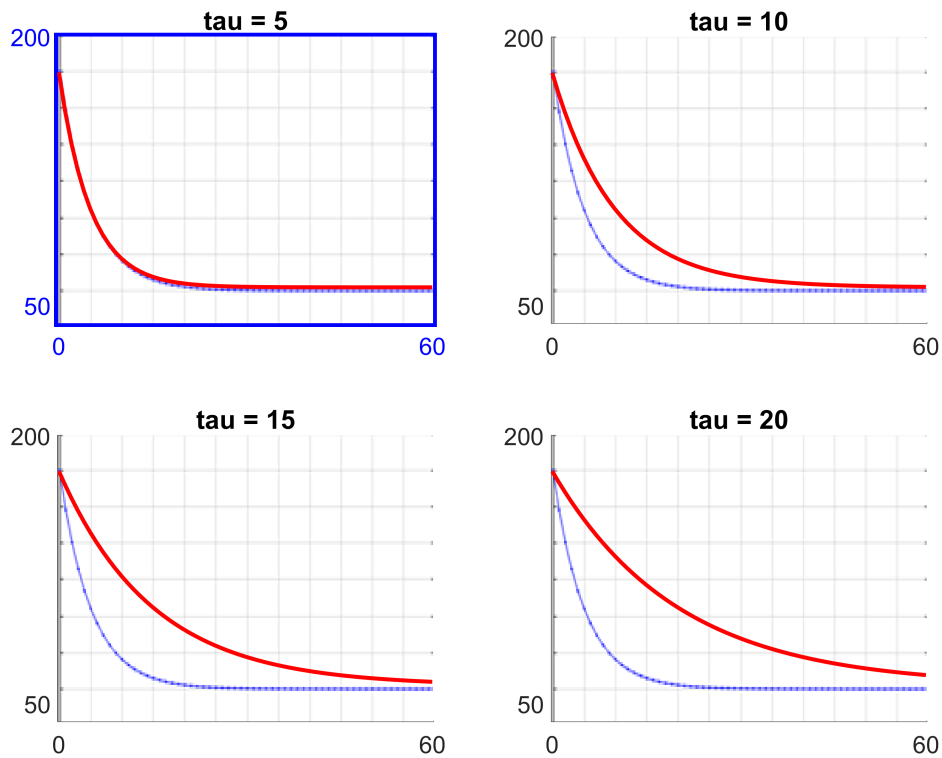
Calculate yt_data, and plot graphs for each potential tau value overlaying figure 2 in a 2 x 2 subplot grid.

```
figure
clf
for i = 1:numel(tau)
 % Calculate heart rate
 yt_data = ca1_fun2([D tau(i) B],t_data);

 % Plot Graphs
 subplot(2,2,i)
```

```
    hold on
    title(sprintf('tau = %i',tau(i)))
    plot(t_data,yt_data,'-r','LineWidth',1.5)
    ylim([40 200])

    % Add graph from fig.2 to the background of the plots
    I = imread('Figure2.png');
    h = image([0 60],[200 40],I);
    uistack(h,'bottom')
    xticks([0 60])
    yticks([50 200])
    hold off
  end
```

The graph tau = 5 matches fig.2 the best out of all values tested.

```
%   Emphasize the plot tau = 5
     ax = subplot(2,2,1);
    box on
    ax.LineWidth = 3;
    ax.XColor = 'blue';
    ax.YColor = 'blue';
```



Generate Figure 2 From The Problem Statement

```
%   Plot Graph
      figure
    ax = axes;
    hold on
```

```
    plot(t_data,ca1_fun2([D tau(1) B],t_data),'-r','LineWidth',1.5);
    ax.XGrid = 'on';
    ax.YGrid = 'on';
    title('Problem Statement Figure 2')
    xlabel('Time (s)')
    ylabel('Heart-rate (BPM)')
    ylim([40 200])
%   Include a text box containing the determined constants
     txt = sprintf('Constants\n\nD = %i BPM \n\\tau = %i s \nB = %i BPM',...
                  D,tau(1),B);
    a = text(45,170,txt);
    a.EdgeColor = 'k';
    a.BackgroundColor = 'white';
    hold off
```