How do unemployment rates and size of labor forces of other US cities compare to those of your city?

By:

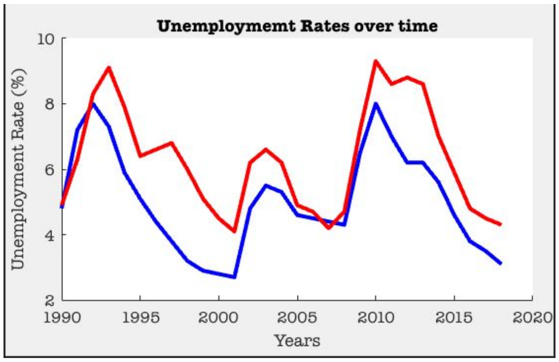
Nafis Abeer

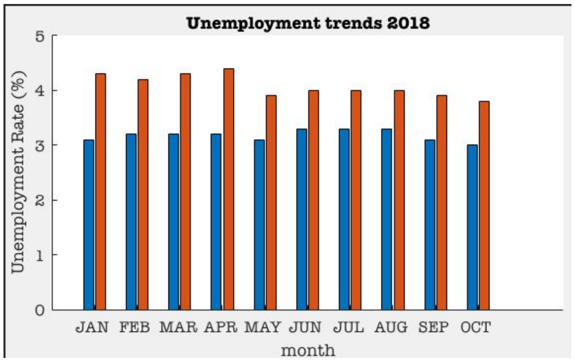
Sohaib Ansari

December 11, 2018

Name of submitted folder:

CityComparisons.mlapp.zip





WE HEREBY CONFIRM THIS IS OUR ORIGINAL WORK:

X______________________ X______________________

Link to unemployment rate data set:

- https://www.bls.gov/web/metro/ssamatab1.txt

App Description:

By making this app, we attempted to provide a better outlook on the employment opportunities in different cities. Every month, the Bureau of Labor Statistics, a sect of The US Department of Labor, records data on employment for every city in the United States. Therefore, we took a dataset that comprised of 395 cities in the U.S. which recorded the unemployment rate, employed workers, and labor force for every month from the year 1990 to 2018.

Our goal of this project was to compare the unemployment rate of two cities and analyze the opportunities for work available in these cities by observing significant trends from the 30 years of data we had. In our program, the user is asked to select two cities. After they are selected, a line graph is shown of both cities' unemployment rates from 1990 to 2018. A double bar graph is shown of the two cities' unemployment rates in 2018 alone. Lastly, a text-box displays significant data comparing the two cities such as notable averages, labor force, predictions and trends.

We believe that our app provides a user proper insight into how specific cities handle their job security through the years. A user maybe looking for work in a different location and our app contributes as a deciding factor to whether a user shall seek jobs in their perspective location in several ways.

The line graph that displays data from 1990 to 2018 allows users to take note of significant peaks and troughs in unemployment rate. The user can compare how the unemployment rate held up during certain years that would give them insight into how strong a city's job security is. For example, a user could observe the unemployment rate during the Great Recession which would provide information on how a city could possibly handle times of economic distress in the future.

The double bar graph that displays 2018 unemployment rates for both cities aid the user in understanding the trends in the current year and where the unemployment rate is headed towards in the future. Our text-box file plays a huge role as our code has a point system that curates a solution to which city would provide a better opportunity for work. Average unemployment rate, labor force, and predicted unemployment rate all garner a point which attributes to which city is better.

The importance of this app is that often times people are stuck in situations where their location of work does not provide enough means of support for them. Our implementation of unemployment rates shows how a city handles its labor force throughout the years and shows significant trends they may be a benefactor to moving to another city for work. Often times cities can increase/decrease in unemployment rate and our app helps predict unemployment rates so that people can be placed in a better position in society than they are today.

File Information:

- Lines 37 - 298, are the lines that are executed once the user presses compare are all reliant on reading from a file and and calling values based on the index of the values (cities) that the user selects from two drop down menus.
- Line 51 loads the data from a matrix file containing the unemployment rates and labor force sizes along with many other relevant data for 395 cities in the US for every month since 1990.

Indexing into data structure:

- Lines 41 - 42 creates a cell array with the names of all the cities in the order that it is written inside the matrix file containing all the data. The cell array is then converted to a character array and assigned to a variable named *cinput*.
- Lines 46 - 47 records the value of the users selections for their city and any US city of their choice in the form of a character vector as well so the values could be compared to an element of *cinput*.
- Lines 56 - 65 initializes variables and assigns them preallocated vectors to store the data that would either be plotted later or written to a file using fprintf to be displayed under the information box.
- Lines 72 - 97 loops through the elements of *cinput* until the index of an element in *cinput* is matched with the values that the user selected and uses the indexes to assign data from the loaded data set to the preallocated variables.
- Lines 101 - 102 takes all the recorded variables which now have data containing information for the user's desired cities and places those data into a struct.

Data Calculations:

- Lines 109 - 110 pre allocates two "points", one for each of the cities. These values go up and down based on whichever city has better numbers in a specific attribute. These values are used to determine which city is better fit to look for jobs in.
- Lines 117 - 118 calculates the average unemployment rate for both the cities for the first month of each year since 1990. Lines 121 - 122 does the same for every month in 2018.
- Lines 125 - 126 calculates the slope of the unemployment rates for the past four months which is used in lines 129 - 130 to make predictions for the unemployment rate of the next month.

Writing to a file:

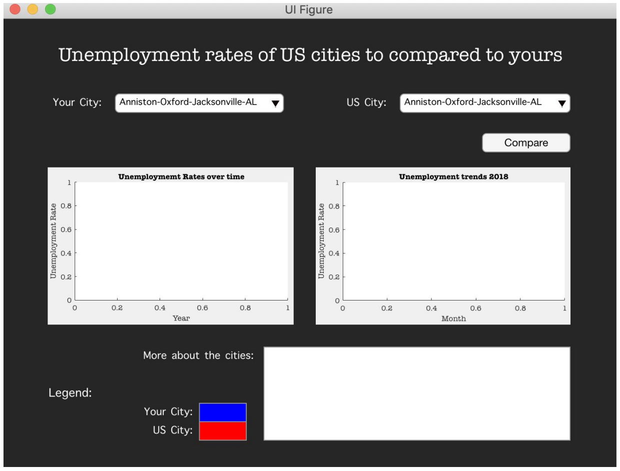
- Line 106 opens a file to be written to and assigns it to a file ID.
- Lines 136 - 214 is basically a deep analysis of the different data points that have been recorded for each of the cities by far.
- Lines 136 - 144 displays which one of the cities have a larger labor force or if they have the same sized labor force. The result is written to a file. The city with the larger labor force receives a point in these lines as well.
- Lines 146 - 156 writes to a file which of the cities have a higher average unemployment rate over the years.
- Line 159 writes the current unemployment rates of both cities to a file. Line 168 writes the average unemployment rates to a file.
- Lines 178 - 193 records the trends of the unemployment rates of both cities to a file. Lines 196 - 197 writes the predictions of both cities for November of 2018 to a file.
- Lines 208 - 214 writes which city, if either, is more suitable to look for work in based on whichever city has more recorded points by this time.
- Line 216 closes the writing file and line 218 reopens the file in reading format.
- Lines 219 - 240 reads a line off from the file ID one at a time and assigns each of the line to a variable. Lines 243 displays all those variables in a textbox using sprintf, so the user could view all the data analysis that have been saved to a file.

Plotting Meaningful Data:

- Lines 252 - 271 plots the unemployment rates of each city for the first month of every year since 1990. These lines assign the graph a title, names the axes and colors the lines based on which city they represent.
- Lines 275 - 286 plots the unemployment trends for each month of 2018 for both cities. The x-axes this time receives months instead of numbers.
- Lines 290 - 291 renames the legends based on the user’s selection of cities.

How to run the app:

Running this app is a rather simple process as we do not have many separate files associated with successfully running the app; just unload the three zip files in the same folder as the MatLab directory on your computer. One of the zip files is the app, one of them contains the data for all the cities and one of them is the file that gets written to when making the textbox. To run the app, the user must unzip the associated .mlapp file and run it. Once the figure to the bottom left is displayed, the user has the option to pick their city and the city they desire to compare the unemployment data to. The user must then press compare and the program runs and displays the unemployment trend of each since 1990 and the unemployment rate trend for each month of 2018. The text box also becomes filled with an analysis of the labor force, average unemployment rate and predictions for both cities. The user is also presented with the programs idea of a better city based on the developed point system. Certain cities could be more fit for a user, based on which attribute, like labor force size of predicted unemployment rate, matter more to the user.



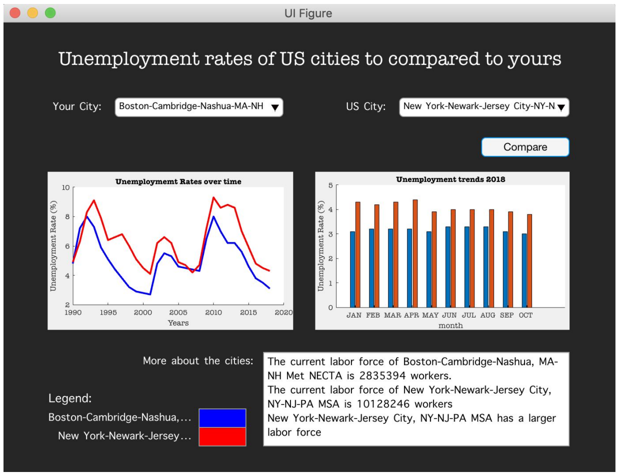
Your City: Anniston-Oxford-Jacksonville-AL

US City: Anniston-Oxford-Jacksonville-AL

More about the cities:

Your City: 🟦

US City: 🟥



Your City: Boston-Cambridge-Nashua-MA-NH

US City: New York-Newark-Jersey City-NY-N

More about the cities: The current labor force of Boston-Cambridge-Nashua, MA-NH Met NECTA is 2835394 workers.
The current labor force of New York-Newark-Jersey City, NY-NJ-PA MSA is 10128246 workers
New York-Newark-Jersey City, NY-NJ-PA MSA has a larger labor force

Boston-Cambridge-Nashua,… 🟦

New York-Newark-Jersey… 🟥

App Designer Code:

```
classdef CityComparisons < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                    matlab.ui.Figure
        YourCityDropDownLabel       matlab.ui.control.Label
        YourCityDropDown            matlab.ui.control.DropDown
        CompareButton               matlab.ui.control.Button
        UIAxes                      matlab.ui.control.UIAxes
        UIAxes2                     matlab.ui.control.UIAxes
        MoreaboutthecitiesLabel     matlab.ui.control.Label
        MoreaboutthecitiesTextArea  matlab.ui.control.TextArea
        UnemploymentratesofUScitiestocomparedtoyoursLabel  matlab.ui.control.Label
        USCityLabel_2               matlab.ui.control.Label
        USCityDropDown              matlab.ui.control.DropDown
        YourCityEditFieldLabel      matlab.ui.control.Label
        YourCityEditField           matlab.ui.control.EditField
        LegendLabel                 matlab.ui.control.Label
        USCityLabel                 matlab.ui.control.Label
        USCityEditField             matlab.ui.control.EditField
    end


    properties (Access = private)
        value;
        value2;
        value3;
        value4;
        value5;

    end


    methods (Access = private)

        % Button pushed function: CompareButton
        function compareButtonPushed(app, event)
       % An array is initialized so it can later be looped through and
       % compared to the values of the user's desired cities

            input = {[name of all the cities]};
            cinput = char(input);

       % The users selection of cities are recorded into variables

            app.value = char(app.YourCityDropDown.Value);
            app.value2 = char(app.USCityDropDown.Value);

       % Open file containing unemployment data for all the cities

           load citydata.mat

       % Initialize column vector to store unemployment rate data for the first month
       % of every year since 1990

            ratevec1 = zeros(29,1);
            ratevec2 = zeros(29,1);

       % Initialize column vector to store unemployment rate data for every month of 2018

            cvec1 = zeros(10,1);
            cvec2 = zeros(10,1);

            lf1 = 0;
            lf2 = 0;
       % For loop that checks if user value is equivilant to any input values
       % and assigns user selections a certain number based on location
       % in input struct

       for i = 1 : 395
           if strcmpi(app.value,input{i})
               % assign values to first month unemployment rates of your city since 1990
               ratevec1 = UnemploymentData.URate(i+3:4740:end);
               % assign values to unemployment rates of every month in 2018 to your city
               cvec1 = UnemploymentData.URate(i+3+132720:395:136673);
               % store names of city
               name1 = UnemploymentData.City(i+3);
               % Find number of people in labor force in october 2018
               lf1= UnemploymentData.LaborForce(i+3+136275);
           end
       end

       for i = 1 : 395

           if strcmpi(app.value2,input{i})
               % assign values to first month unemployment rates of US city since 1990
               ratevec2 = UnemploymentData.URate(i+3:4740:end);
               % assign values to unemployment rates of every month in 2018 to US city
               cvec2 = UnemploymentData.URate(i+3+132720:395:136673);
               % store names of city
               name2 = UnemploymentData.City(i+3);
               % Find number of people in labor force in october 2018
               lf2= UnemploymentData.LaborForce(i+3+136275);
           end
       end

       % Places values in a data structure

            CityUR(2) = struct('Name', name2, 'URates', ratevec2, 'CurrentRate', cvec2,
'LaborForce', lf2);
            CityUR(1) = struct('Name', name1,'URates', ratevec1, 'CurrentRate', cvec1,
'LaborForce', lf1);

       % Analyze values to intialize a file that will be used to write to text box

        fid = fopen('CityAnalysis.rtf','w');

        %Point System To Determine Which City Is Best To Work
        C1pt=0;
        C2pt=0;

        %Gets Current Unemployment Rate
        C1C = CityUR(1).CurrentRate(10);
        C2C = CityUR(2).CurrentRate(10);

        %Gets Average From 1990-2018
        C1Avg = mean(CityUR(1).URat11les);
        C2Avg = mean(CityUR(2).URates);

        %Gets Average For 2018
        C1Avg18 = mean(CityUR(1).CurrentRate);
        C2Avg18 = mean(CityUR(2).CurrentRate);

        %Slope For Last 4 Months
        C1Slope = (CityUR(1).CurrentRate(10) - CityUR(1).CurrentRate(7))/4;
        C2Slope = (CityUR(2).CurrentRate(10) - CityUR(2).CurrentRate(7))/4;

        %Prediction For November
        C1Pre = (C1Slope * 5) + CityUR(1).CurrentRate(7);
        C2Pre = (C2Slope * 5) + CityUR(2).CurrentRate(7);

        %Writes Analysis Into A File
        fprintf(fid, 'The current labor force of %s is %d workers.\nThe current labor force of
%s is %d workers\n', CityUR(1).Name, CityUR(1).LaborForce, CityUR(2).Name,
CityUR(2).LaborForce);

        %Writes and compares labor force
        if CityUR(1).LaborForce > CityUR(2).LaborForce
            fprintf(fid, '%s has a larger labor force\n\n',CityUR(1).Name);
            C1pt= C1pt+1;
        elseif CityUR(1).LaborForce < CityUR(2).LaborForce
            fprintf(fid, '%s has a larger labor force\n\n',CityUR(2).Name);
            C2pt= C2pt+1;
        else
            fprintf(fid, 'Both cities have the same amount of workers!\n\n');
        end

        %Writes and compares avg unemployment from 1990-2018
        fprintf(fid, 'From 1990-2018 the average unemployment rate of %s is %.2f%%.\nAnd the
average unemployment rate of %s is %.2f%%\n', CityUR(1).Name, C1Avg, CityUR(2).Name, C2Avg);
        if C1Avg > C2Avg
            fprintf(fid, '%s historically has a smaller unemployment
rate\n\n',CityUR(2).Name);
            C2pt= C2pt+1;
        elseif C1Avg < C2Avg
            fprintf(fid, '%s historically has a smaller unemployment
rate\n\n',CityUR(1).Name);
            C1pt= C1pt+1;
        else
            fprintf(fid, 'Both cities have the same average unemployment rates!\n\n');
        end

        %Writes the current unemployment rates
        fprintf(fid, 'The current unemployment rate of %s is %.2f%%.\nThe current unemployment
rate of %s is %.2f%%.\n\n', CityUR(1).Name, C1C, CityUR(2).Name, C2C);
        if C1C > C2C
            C2pt= C2pt+1;
        elseif C1C < C2C
            C1pt= C1pt+1;
        else
        end

        %Writes the average unemployment rates
        fprintf(fid, 'The average unemployment rate of %s for 2018 is %.2f%%\nAnd the average
unemployment rate of %s is %.2f%%.\n\n', CityUR(1).Name, C1Avg18, CityUR(2).Name, C2Avg18);
        if C1Avg18 > C2Avg18
            C2pt= C2pt+1;
        elseif C1Avg18 < C2Avg18
            C1pt= C1pt+1;
        else
        end

        %Writes the trend of the unemployment rate for each city in 2018 in the last 4
        %months
        if C1Slope < 0
            fprintf(fid,'%s is at a downward trend for unemployment rate\n', CityUR(1).Name);
            C1pt= C1pt+1;
        elseif C1Slope > 0
            fprintf(fid,'%s is unfortunately at an increasing trend for unemployment rate\n',
CityUR(1).Name);
        else
            fprintf(fid,'%s is at a stagnant unemployment rate\n', CityUR(1).Name)
        end
        if C2Slope < 0
```

```
            fprintf(fid,'%s is at a downward trend for unemployment rate\n\n',
CityUR(2).Name);
            C2pt= C2pt+1;
        elseif C2Slope > 0
            fprintf(fid,'%s is unfortunately at an increasing trend for unemployment
rate\n\n', CityUR(2).Name);
        else
            fprintf(fid,'%s is at a stagnant unemployment rate\n\n', CityUR(2).Name);
        end

        %Writes the predicted unemployment rate for November 2018
        fprintf(fid, 'The predicted unemployment rate for %s in November 2018 is %.2f%%\n',
CityUR(1).Name, C1Pre);
        fprintf(fid, 'The predicted unemployment rate for %s in November 2018 is
%.2f%%\n\n\n', CityUR(2).Name, C2Pre);

        if C1Pre > C2Pre
            C2pt= C2pt+2;
        elseif C1Pre < C2Pre
            C1pt= C1pt+2;
        else
        end


        %Writes which city would provide best opportunity for work
        if C1pt > C2pt
            fprintf(fid, '%s would provide more opportunities for work!', CityUR(1).Name);
        elseif C1pt < C2pt
            fprintf(fid, '%s would provide more opportunities for work!', CityUR(2).Name);
        else
            fprintf(fid, 'Both %s and %s have their pros and cons for opportunites for work',
CityUR(1).Name, CityUR(2).Name);
        end

        fclose('all');

        fid = fopen('CityAnalysis.rtf','r');
        line1 = fgets(fid);
        line2 = fgets(fid);
        line3 = fgets(fid);
        line4 = fgets(fid);
        line5 = fgets(fid);
        line6 = fgets(fid);
        line7 = fgets(fid);
        line8 = fgets(fid);
        line9 = fgets(fid);
        line10 = fgets(fid);
        line11 = fgets(fid);
        line12 = fgets(fid);
        line13 = fgets(fid);
        line14 = fgets(fid);
        line15 = fgets(fid);
        line16 = fgets(fid);
        line17 = fgets(fid);
        line18 = fgets(fid);
        line19 = fgets(fid);
        line20 = fgets(fid);
        line21 = fgets(fid);
        line22 = fgets(fid);


        app.MoreaboutthecitiesTextArea.Value =
sprintf('%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s',
line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line13,line14,line1
5,line16,line17,line18,line19,line20,line21,line22);
```

```
        fclose('all');




       % Plot unemployment trends for first month of both cities since 1990 in UIAxes

            x = 1990:2018;
            y1 = CityUR(1).URates;
            y2 = CityUR(2).URates;


            line1 = plot(app.UIAxes,x,y1);
            line1.Color = [0 0 1];
            line1.LineWidth = 2;

            hold(app.UIAxes,'on');

            line2 = plot(app.UIAxes,x,y2);
            line2.Color = [1 0 0];
            line2.LineWidth = 2;

            xlabel(app.UIAxes,'Years')
            ylabel(app.UIAxes,'Unemployment Rate (%)')


            hold(app.UIAxes,'off');

       % Plot unemployment trends for 2018 of both cities in UIAxes2

            y1 = CityUR(1).CurrentRate;
            y2 = CityUR(2).CurrentRate;


            b = bar(app.UIAxes2,[y1,y2], 'grouped');

            month = {'JAN ' , 'FEB ', 'MAR ' , 'APR ' , 'MAY ' , 'JUN ', 'JUL ' , 'AUG ', 'SEP
', 'OCT '};
            set(app.UIAxes2,'xtick',[1:10],'xticklabel',month);


            xlabel(app.UIAxes2,'month')
            ylabel(app.UIAxes2,'Unemployment Rate (%)')


        % Name the cities in the legend
        app.YourCityEditFieldLabel.Text = sprintf('%s',CityUR(1).Name);
        app.USCityLabel.Text = sprintf('%s', CityUR(2).Name);







        end

        % Value changed function: YourCityDropDown
        function YourCityDropDownValueChanged(app, event)
            app.value = app.YourCityDropDown.Value;

        end

        % Value changed function: USCityDropDown
        function USCityDropDownValueChanged(app, event)
            app.value2 = app.USCityDropDown.Value;
        end

        % Value changed function: MoreaboutthecitiesTextArea
        function MoreaboutthecitiesTextAreaValueChanged(app, event)
            app.value3 = app.MoreaboutthecitiesTextArea.Value;

        end

        % Value changed function: YourCityEditField
        function YourCityEditFieldValueChanged(app, event)
            app.value4 = app.YourCityEditField.Value;

        end
    end

    % App initialization and construction
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure
            app.UIFigure = uifigure;
            app.UIFigure.Color = [0.149 0.149 0.149];
            app.UIFigure.Position = [100 100 694 517];
            app.UIFigure.Name = 'UI Figure';

            % Create YourCityDropDownLabel
            app.YourCityDropDownLabel = uilabel(app.UIFigure);
            app.YourCityDropDownLabel.HorizontalAlignment = 'right';
            app.YourCityDropDownLabel.FontName = 'Apple Symbols';
            app.YourCityDropDownLabel.FontSize = 16;
            app.YourCityDropDownLabel.FontColor = [1 1 1];
            app.YourCityDropDownLabel.Position = [52 411 61 22];
            app.YourCityDropDownLabel.Text = 'Your City:';

            % Create YourCityDropDown
            app.YourCityDropDown = uidropdown(app.UIFigure);
            app.YourCityDropDown.Items = {[name of all the cities]};
            app.YourCityDropDown.ValueChangedFcn = createCallbackFcn(app,
@YourCityDropDownValueChanged, true);
            app.YourCityDropDown.FontName = 'Apple Symbols';
            app.YourCityDropDown.FontSize = 14;
            app.YourCityDropDown.BackgroundColor = [1 1 1];
            app.YourCityDropDown.Position = [128 411 191 22];
            app.YourCityDropDown.Value = 'Anniston-Oxford-Jacksonville-AL';

            % Create CompareButton
            app.CompareButton = uibutton(app.UIFigure, 'push');
            app.CompareButton.ButtonPushedFcn = createCallbackFcn(app, @compareButtonPushed,
true);
            app.CompareButton.Position = [543 366 100 22];
            app.CompareButton.Text = 'Compare';

            % Create UIAxes
            app.UIAxes = uiaxes(app.UIFigure);
            title(app.UIAxes, 'Unemploymemt Rates over time')
            xlabel(app.UIAxes, 'Year')
            ylabel(app.UIAxes, 'Unemployment Rate')
            app.UIAxes.PlotBoxAspectRatio = [1 0.557203389830508 0.557203389830508];
            app.UIAxes.FontName = 'American Typewriter';
            app.UIAxes.FontSize = 8;
            app.UIAxes.Position = [52 170 278 179];

            % Create UIAxes2
            app.UIAxes2 = uiaxes(app.UIFigure);
            title(app.UIAxes2, 'Unemployment trends 2018')
            xlabel(app.UIAxes2, 'Month')
            ylabel(app.UIAxes2, 'Unemployment Rate')
            app.UIAxes2.PlotBoxAspectRatio = [1 0.534552845528455 0.534552845528455];
            app.UIAxes2.FontName = 'American Typewriter';
            app.UIAxes2.FontSize = 8;
            app.UIAxes2.Position = [355 170 288 179];

            % Create MoreaboutthecitiesLabel
            app.MoreaboutthecitiesLabel = uilabel(app.UIFigure);
            app.MoreaboutthecitiesLabel.BackgroundColor = [0.149 0.149 0.149];
            app.MoreaboutthecitiesLabel.HorizontalAlignment = 'right';
            app.MoreaboutthecitiesLabel.FontName = 'Apple Symbols';
            app.MoreaboutthecitiesLabel.FontSize = 16;
            app.MoreaboutthecitiesLabel.FontColor = [1 1 1];
            app.MoreaboutthecitiesLabel.Position = [154 123 131 22];
            app.MoreaboutthecitiesLabel.Text = 'More about the cities:';

            % Create MoreaboutthecitiesTextArea
            app.MoreaboutthecitiesTextArea = uitextarea(app.UIFigure);
            app.MoreaboutthecitiesTextArea.ValueChangedFcn = createCallbackFcn(app,
@MoreaboutthecitiesTextAreaValueChanged, true);
            app.MoreaboutthecitiesTextArea.FontName = 'Apple Symbols';
            app.MoreaboutthecitiesTextArea.FontSize = 16;
            app.MoreaboutthecitiesTextArea.Position = [296 38 347 107];

            % Create UnemploymentratesofUScitiestocomparedtoyoursLabel
            app.UnemploymentratesofUScitiestocomparedtoyoursLabel = uilabel(app.UIFigure);
            app.UnemploymentratesofUScitiestocomparedtoyoursLabel.HorizontalAlignment =
'center';
            app.UnemploymentratesofUScitiestocomparedtoyoursLabel.VerticalAlignment = 'top';
            app.UnemploymentratesofUScitiestocomparedtoyoursLabel.FontName = 'American
Typewriter';
            app.UnemploymentratesofUScitiestocomparedtoyoursLabel.FontSize = 22;
            app.UnemploymentratesofUScitiestocomparedtoyoursLabel.FontColor = [1 1 1];
            app.UnemploymentratesofUScitiestocomparedtoyoursLabel.Position = [-280 452 1258
40];
            app.UnemploymentratesofUScitiestocomparedtoyoursLabel.Text = 'Unemployment rates
of US cities to compared to yours';

            % Create USCityLabel_2
            app.USCityLabel_2 = uilabel(app.UIFigure);
            app.USCityLabel_2.HorizontalAlignment = 'right';
            app.USCityLabel_2.FontName = 'Apple Symbols';
            app.USCityLabel_2.FontSize = 16;
            app.USCityLabel_2.FontColor = [1 1 1];
            app.USCityLabel_2.Position = [385 411 50 22];
            app.USCityLabel_2.Text = 'US City:';

            % Create USCityDropDown
            app.USCityDropDown = uidropdown(app.UIFigure);
            app.USCityDropDown.Items = {[name of all the cities]};
            app.USCityDropDown.ValueChangedFcn = createCallbackFcn(app,
@USCityDropDownValueChanged, true);
            app.USCityDropDown.FontName = 'Apple Symbols';
            app.USCityDropDown.FontSize = 14;
            app.USCityDropDown.BackgroundColor = [1 1 1];
            app.USCityDropDown.Position = [450 411 193 22];
            app.USCityDropDown.Value = 'Anniston-Oxford-Jacksonville-AL';

            % Create YourCityEditFieldLabel
            app.YourCityEditFieldLabel = uilabel(app.UIFigure);
            app.YourCityEditFieldLabel.HorizontalAlignment = 'right';
            app.YourCityEditFieldLabel.FontName = 'Apple Symbols';
            app.YourCityEditFieldLabel.FontSize = 16;
            app.YourCityEditFieldLabel.FontColor = [1 1 1];
            app.YourCityEditFieldLabel.Position = [48 59 168 22];
            app.YourCityEditFieldLabel.Text = 'Your City:';

            % Create YourCityEditField
            app.YourCityEditField = uieditfield(app.UIFigure, 'text');
            app.YourCityEditField.ValueChangedFcn = createCallbackFcn(app,
@YourCityEditFieldValueChanged, true);
            app.YourCityEditField.FontName = 'Apple Symbols';
            app.YourCityEditField.FontSize = 16;
            app.YourCityEditField.BackgroundColor = [0 0 1];
            app.YourCityEditField.Position = [223 59 54 22];

            % Create LegendLabel
            app.LegendLabel = uilabel(app.UIFigure);
            app.LegendLabel.HorizontalAlignment = 'right';
            app.LegendLabel.FontName = 'Apple Symbols';
            app.LegendLabel.FontSize = 18;
            app.LegendLabel.FontColor = [1 1 1];
            app.LegendLabel.Position = [48 80 54 22];
            app.LegendLabel.Text = 'Legend:';

            % Create USCityLabel
            app.USCityLabel = uilabel(app.UIFigure);
            app.USCityLabel.HorizontalAlignment = 'right';
            app.USCityLabel.FontName = 'Apple Symbols';
            app.USCityLabel.FontSize = 16;
            app.USCityLabel.FontColor = [1 1 1];
            app.USCityLabel.Position = [59 38 157 22];
            app.USCityLabel.Text = 'US City:';

            % Create USCityEditField
            app.USCityEditField = uieditfield(app.UIFigure, 'text');
            app.USCityEditField.FontName = 'Apple Symbols';
            app.USCityEditField.FontSize = 16;
            app.USCityEditField.FontColor = [1 1 1];
            app.USCityEditField.BackgroundColor = [1 0 0];
            app.USCityEditField.Position = [223 38 54 22];
        end
    end

    methods (Access = public)

        % Construct app
        function app = CityComparisons

            % Create and configure components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```