```
classdef sealevels < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure             matlab.ui.Figure
        TimeDropDownLabel    matlab.ui.control.Label
        TimeDropDown         matlab.ui.control.DropDown
        UIAxes               matlab.ui.control.UIAxes
        AllButton            matlab.ui.control.StateButton
        AntarcticTempButton  matlab.ui.control.StateButton
        SeaLevelButton       matlab.ui.control.StateButton
        CO2LevelButton       matlab.ui.control.StateButton
        GlacierMassButton    matlab.ui.control.StateButton
        LOADDATAButton       matlab.ui.control.Button
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Callback function
        function SalinityButtonValueChanged(app, event)

        end

        % Value changed function: CO2LevelButton
        function CO2LevelButtonValueChanged(app, event)
            %turn other buttons off
            app.GlacierMassButton.Value = 0;
            app.SeaLevelButton.Value = 0;
            app.AllButton.Value = 0;
            app.AntarcticTempButton.Value = 0;

            value = app.CO2LevelButton.Value;
            %load carbon dioxide data
            load whyphy.dat
            x = whyphy(:,1);
            y = whyphy(:,2);
            if value == 1
                %configure plot and axes steps/labels
                cla(app.UIAxes,'reset');
                plot(app.UIAxes,x,y);
                app.UIAxes.XTick = 1985:5:2020;
                app.UIAxes.XTickLabel = 1985:5:2020;
                app.UIAxes.XLabel.String = 'Year';
                app.UIAxes.YTick = 350:10:420;
                app.UIAxes.YTickLabel = 350:10:420;
                app.UIAxes.YLabel.String = 'CO2 Concentration (parts
 per million)';
                app.UIAxes.Title.String = 'Carbon Dioxide
 Concentration in Atmosphere since 1989';
                legend(app.UIAxes,'hide');
            elseif value == 0
```

```
                %clear data and axes
                cla(app.UIAxes,'reset');
                app.UIAxes.XTick = '';
                app.UIAxes.XTickLabel = '';
                app.UIAxes.YTick = '';
                app.UIAxes.YTickLabel = '';
            end
        end

        % Button pushed function: LOADDATAButton
        function LOADDATAButtonPushed(app, event)
            %calls functions to put data sets into preferred formats
            nhtempdata();   %temperature
            co2();          %carbon dioxide
            sealeveldata(); %sea level
            glacierdata();  %glacier mass

            %don't need this button again
            app.LOADDATAButton.Visible = 0;
            app.LOADDATAButton.Enable = 0;
        end

        % Value changed function: SeaLevelButton
        function SeaLevelButtonValueChanged(app, event)
            %turn other buttons off
            app.GlacierMassButton.Value = 0;
            app.CO2LevelButton.Value = 0;
            app.AllButton.Value = 0;
            app.AntarcticTempButton.Value = 0;

            value = app.SeaLevelButton.Value;
            %load sea level data
            load mynamejeff.dat
            x = mynamejeff(:,1);
            y = mynamejeff(:,2);
            if value == 1
                %configure plot and axes steps/labels
                cla(app.UIAxes,'reset');
                plot(app.UIAxes,x,y);
                app.UIAxes.XTick = 1990:5:2020;
                app.UIAxes.XTickLabel = 1990:5:2020;
                app.UIAxes.XLabel.String = 'Year';
                app.UIAxes.YTick = -10:10:90;
                app.UIAxes.YTickLabel = -10:10:90;
                app.UIAxes.YLabel.String = 'Sea Level (centimeters)';
                app.UIAxes.Title.String = 'Rise in Global Mean Sea
Level since 1993';
                legend(app.UIAxes,'hide');
            elseif value == 0
                %clear data and axes
                cla(app.UIAxes,'reset');
                app.UIAxes.XTick = '';
                app.UIAxes.XTickLabel = '';
                app.UIAxes.YTick = '';
```

```
                app.UIAxes.YTickLabel = '';
            end
        end

        % Value changed function: AntarcticTempButton
        function AntarcticTempButtonValueChanged(app, event)
            %turn off other buttons
            app.GlacierMassButton.Value = 0;
            app.CO2LevelButton.Value = 0;
            app.AllButton.Value = 0;
            app.SeaLevelButton.Value = 0;

            value = app.AntarcticTempButton.Value;
            %load temperature data
            load somethingcool.dat
            x = somethingcool(:,1);
            y = somethingcool(:,2);
            if value == 1
                %configure plot and axes steps/labels
                cla(app.UIAxes,'reset');
                plot(app.UIAxes,x,y);
                app.UIAxes.XTick = 1840:10:2020;
                app.UIAxes.XTickLabel = 1840:10:2020;
                app.UIAxes.XLabel.String = 'Year';
                app.UIAxes.YTick = -1:.1:1.5;
                app.UIAxes.YTickLabel = -1:.1:1.5;
                app.UIAxes.YLabel.String = 'Temperature (°Celsius)';
                app.UIAxes.Title.String = 'Rise in Average Temperature
of Northern Hemisphere since 1850';
                legend(app.UIAxes,'hide');
            elseif value == 0
                %clear data and axes
                cla(app.UIAxes,'reset');
                app.UIAxes.XTick = '';
                app.UIAxes.XTickLabel = '';
                app.UIAxes.YTick = '';
                app.UIAxes.YTickLabel = '';
            end
        end

        % Value changed function: GlacierMassButton
        function GlacierMassButtonValueChanged(app, event)
            %turn off other buttons
            app.CO2LevelButton.Value = 0;
            app.SeaLevelButton.Value = 0;
            app.AllButton.Value = 0;
            app.AntarcticTempButton.Value = 0;

            value = app.GlacierMassButton.Value;
            %load glacier melt data
            load meatqute.dat
            x = meatqute(:,1);
            y = meatqute(:,2);
            if value == 1
```

```
                %configure plot and axes steps/labels
                cla(app.UIAxes,'reset');
                plot(app.UIAxes,x,y);
                app.UIAxes.XTick = 2000:2:2020;
                app.UIAxes.XTickLabel = 2000:2:2020;
                app.UIAxes.XLabel.String = 'Year';
                app.UIAxes.YTick = -2000:200:400;
                app.UIAxes.YTickLabel = -2000:200:400;
                app.UIAxes.YLabel.String = 'Glacier Mass
(Gigatonnes)';
                app.UIAxes.Title.String = 'Antarctic Mass Variation
since 2002';
                legend(app.UIAxes,'hide');
            elseif value == 0
                %clear data and axes
                cla(app.UIAxes,'reset');
                app.UIAxes.XTick = '';
                app.UIAxes.XTickLabel = '';
                app.UIAxes.YTick = '';
                app.UIAxes.YTickLabel = '';
            end
        end

        % Value changed function: TimeDropDown
        function TimeDropDownValueChanged(app, event)
            value = app.TimeDropDown.Value;
            if value == app.TimeDropDown.Items{1} %past
                %make buttons visible, enable use
                app.CO2LevelButton.Visible = 1;
                app.SeaLevelButton.Enable = 1;
                app.SeaLevelButton.Visible = 1;
                app.CO2LevelButton.Enable = 1;
                app.AntarcticTempButton.Visible = 1;
                app.AntarcticTempButton.Enable = 1;
                app.AllButton.Visible = 1;
                app.AllButton.Enable = 1;
                app.GlacierMassButton.Visible = 1;
                app.GlacierMassButton.Enable = 1;
                app.UIAxes.Title.String = '';
                %clear data and axes
                cla(app.UIAxes,'reset');
                app.UIAxes.XTick = '';
                app.UIAxes.XTickLabel = '';
                app.UIAxes.YTick = '';
                app.UIAxes.YTickLabel = '';
            elseif value == app.TimeDropDown.Items{2} %future
                %set all buttons invisible, disable
                %only want to graph future sea level data
                app.CO2LevelButton.Value = 0;
                app.CO2LevelButton.Enable = 0;
                app.CO2LevelButton.Visible = 0;
                app.SeaLevelButton.Value = 0;
                app.SeaLevelButton.Enable = 0;
                app.SeaLevelButton.Visible = 0;
```

```
                app.AllButton.Value = 0;
                app.AllButton.Enable = 0;
                app.AllButton.Visible = 0;
                app.AntarcticTempButton.Value = 0;
                app.AntarcticTempButton.Enable = 0;
                app.AntarcticTempButton.Visible = 0;
                app.GlacierMassButton.Value = 0;
                app.GlacierMassButton.Enable = 0;
                app.GlacierMassButton.Visible = 0;
                cla(app.UIAxes,'reset');
                legend(app.UIAxes,'hide');

                %loads approximation data of future sea level
                load FutureSL2.dat
                x = FutureSL2(:,1);
                y = FutureSL2(:,2);
                plot(app.UIAxes,x,y);

                %estimates for when specific cities could be
        underwater
                xconst = 2015:2100;
                xlen = length(xconst);
                shanghai = ones(xlen)*338;
                miami = ones(xlen)*200;
                bangkok = ones(xlen)*150;
                dhaka = ones(xlen)*348;
                galveston = ones(xlen)*213.36;
                nola = ones(xlen)*180;
                nyc = ones(xlen)*300;
                hold(app.UIAxes);
                plot(app.UIAxes,xconst,shanghai);
                text(app.UIAxes,2015,shanghai(1)-4,'Shanghai');
                plot(app.UIAxes,xconst,miami);
                text(app.UIAxes,2015,miami(1)-4,'Miami, FL');
                plot(app.UIAxes,xconst,bangkok);
                text(app.UIAxes,2015,bangkok(1)-4,'Bangkok');
                plot(app.UIAxes,xconst,dhaka);
                text(app.UIAxes,2015,dhaka(1)-4,'Dhaka');
                plot(app.UIAxes,xconst,galveston);
                text(app.UIAxes,2015,galveston(1)-4,'Galveston, TX');
                plot(app.UIAxes,xconst,nola);
                text(app.UIAxes,2015,nola(1)-4,'New Orleans');
                plot(app.UIAxes,xconst,nyc);
                text(app.UIAxes,2015,nyc(1)-4,'New York City');
                hold(app.UIAxes);

                %set axes steps and labels
                app.UIAxes.XTick = 2015:5:2100;
                app.UIAxes.XTickLabel = 2015:5:2100;
                app.UIAxes.XLabel.String = 'Year';
                app.UIAxes.YTick = 70:20:350;
                app.UIAxes.YTickLabel = 70:20:350;
                app.UIAxes.YLabel.String = 'Sea Level (centimeters)';
```

```
                app.UIAxes.Title.String = 'Approximated Sea Level and
When Cities Will be Underwater';
            end
        end

        % Value changed function: AllButton
        function AllButtonValueChanged(app, event)
            %turn other buttons off
            app.CO2LevelButton.Value = 0;
            app.SeaLevelButton.Value = 0;
            app.GlacierMassButton.Value = 0;
            app.AntarcticTempButton.Value = 0;

            value = app.AllButton.Value;
            if value == 1
                %adjust data--scale down so all four data sets can be
seen and interpreted in the same plot
                %glacier mass data
                load meatqute.dat
                xice = meatqute(:,1);
                yice = meatqute(:,2);
                y1 = -log10(yice) + 7;
                maticemass(:,1) = xice;
                maticemass(:,2) = y1;

                %temperature data
                load somethingcool.dat
                xt = somethingcool(:,1);
                yt = somethingcool(:,2);
                %only 2002-present
                xtemp = xt(153:end);
                ytemp = yt(153:end);
                y2 = 5 * abs(ytemp) + 1;
                matnhtemp(:,1) = xtemp;
                matnhtemp(:,2) = y2;

                %carbon dioxide data
                load whyphy.dat
                xc = whyphy(:,1);
                yc = whyphy(:,2);
                %only 2002-present
                xco2 = xc(157:end);
                yco2 = yc(157:end);
                y3 = (abs(yco2))/100;
                matCO2(:,1) = xco2;
                matCO2(:,2) = y3;

                %sea level data
                load mynamejeff.dat
                xs = mynamejeff(:,1);
                ys = mynamejeff(:,2);
                %only 2002-present
                xsea = xs(108:end);
                ysea = ys(108:end);
```

```
                y4 = (abs(ysea))/10;
                matsealevel(:,1) = xsea;
                matsealevel(:,2) = y4;

                cla(app.UIAxes,'reset');

plot(app.UIAxes,maticemass(:,1),maticemass(:,2),matsealevel(:,1),matsealevel(:,2)
                legend(app.UIAxes,'Ice Mass (Gt)','NH Temp. (°C)','CO2
Conc. (ppm)','Sea Level (cm)','Location','northwest');

                %set axes steps and labels
                app.UIAxes.XTick = 2000:2:2020;
                app.UIAxes.XTickLabel = 2000:2:2020;
                app.UIAxes.XLabel.String = 'Year';
                app.UIAxes.YTick = '';
                app.UIAxes.YTickLabel = '';
                app.UIAxes.YLabel.String = '';
                app.UIAxes.Title.String = 'Data Correlation';
            elseif value == 0
                %clear data and axes
                cla(app.UIAxes,'reset');
                app.UIAxes.XTick = '';
                app.UIAxes.XTickLabel = '';
                app.UIAxes.YTick = '';
                app.UIAxes.YTickLabel = '';
                %legend(app.UIAxes,'hide');
            end
        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are
created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Color = [0 0 0];
            app.UIFigure.Position = [100 100 640 480];
            app.UIFigure.Name = 'UI Figure';

            % Create TimeDropDownLabel
            app.TimeDropDownLabel = uilabel(app.UIFigure);
            app.TimeDropDownLabel.HorizontalAlignment = 'right';
            app.TimeDropDownLabel.FontColor = [1 1 1];
            app.TimeDropDownLabel.Position = [17 445 32 22];
            app.TimeDropDownLabel.Text = 'Time';

            % Create TimeDropDown
            app.TimeDropDown = uidropdown(app.UIFigure);
            app.TimeDropDown.Items = {'Past  ', 'Future'};
```

```
            app.TimeDropDown.ValueChangedFcn = createCallbackFcn(app,
@TimeDropDownValueChanged, true);
            app.TimeDropDown.Position = [64 445 67 22];
            app.TimeDropDown.Value = 'Past  ';

            % Create UIAxes
            app.UIAxes = uiaxes(app.UIFigure);
            title(app.UIAxes, '')
            app.UIAxes.PlotBoxAspectRatio = [1.52845528455285 1 1];
            app.UIAxes.XTick = [];
            app.UIAxes.XTickLabel = '';
            app.UIAxes.YTick = [];
            app.UIAxes.YTickLabel = '';
            app.UIAxes.Position = [17 10 613 425];

            % Create AllButton
            app.AllButton = uibutton(app.UIFigure, 'state');
            app.AllButton.ValueChangedFcn = createCallbackFcn(app,
@AllButtonValueChanged, true);
            app.AllButton.Text = 'All';
            app.AllButton.BackgroundColor = [0.3922 0.8314 0.0745];
            app.AllButton.Position = [596 445 34 22];

            % Create AntarcticTempButton
            app.AntarcticTempButton = uibutton(app.UIFigure, 'state');
            app.AntarcticTempButton.ValueChangedFcn =
createCallbackFcn(app, @AntarcticTempButtonValueChanged, true);
            app.AntarcticTempButton.Text = 'Antarctic Temp';
            app.AntarcticTempButton.BackgroundColor = [1 1 1];
            app.AntarcticTempButton.Position = [391.5 445 96 22];

            % Create SeaLevelButton
            app.SeaLevelButton = uibutton(app.UIFigure, 'state');
            app.SeaLevelButton.ValueChangedFcn =
createCallbackFcn(app, @SeaLevelButtonValueChanged, true);
            app.SeaLevelButton.Text = 'Sea Level';
            app.SeaLevelButton.BackgroundColor = [0.0706 0.6196 1];
            app.SeaLevelButton.Position = [231 445 66 22];

            % Create CO2LevelButton
            app.CO2LevelButton = uibutton(app.UIFigure, 'state');
            app.CO2LevelButton.ValueChangedFcn =
createCallbackFcn(app, @CO2LevelButtonValueChanged, true);
            app.CO2LevelButton.Text = 'CO2 Level';
            app.CO2LevelButton.BackgroundColor = [1 1 0];
            app.CO2LevelButton.Position = [309 445 72 22];

            % Create GlacierMassButton
            app.GlacierMassButton = uibutton(app.UIFigure, 'state');
            app.GlacierMassButton.ValueChangedFcn =
createCallbackFcn(app, @GlacierMassButtonValueChanged, true);
            app.GlacierMassButton.Text = 'Glacier Mass';
            app.GlacierMassButton.BackgroundColor = [0 1 1];
            app.GlacierMassButton.Position = [498 445 89 22];
```

```
            % Create LOADDATAButton
            app.LOADDATAButton = uibutton(app.UIFigure, 'push');
            app.LOADDATAButton.ButtonPushedFcn =
 createCallbackFcn(app, @LOADDATAButtonPushed, true);
            app.LOADDATAButton.BackgroundColor = [1 0 0];
            app.LOADDATAButton.Position = [143 445 75 22];
            app.LOADDATAButton.Text = 'LOAD DATA';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = sealevels

            % Create UIFigure and 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
```

Published with MATLAB® R2019a