

Graphic User Interface - App Designer

MATLAB進階程式語言與實作

盧家鋒 Chia-Feng Lu, Ph.D.
Department of Biomedical Imaging and
Radiological Sciences, NYCU
alvin4016@nycu.edu.tw

Teaching Materials

<http://cflu.lab.nycu.edu.tw>

Contents → Teaching Materials → MATLAB ML (G)

Please download **Week 9** Materials. Compulsory Course for the Undergraduate Students

Lecturer: Chia-Feng Lu (alvin4016@ym.edu.tw)

Matlab進階程式設計與專題實作 (碩博)

授課教師：盧家鋒

Please set current directory to **MLmaterials_L9**

Home Contents

MATLAB Programming for Machine Learning (Graduate)

Compulsory Course for the Undergraduate Students



Lecturer: Chia-Feng Lu (alvin4016@ym.edu.tw)

Matlab進階程式設計與專題實作 (碩博)

授課教師：盧家鋒

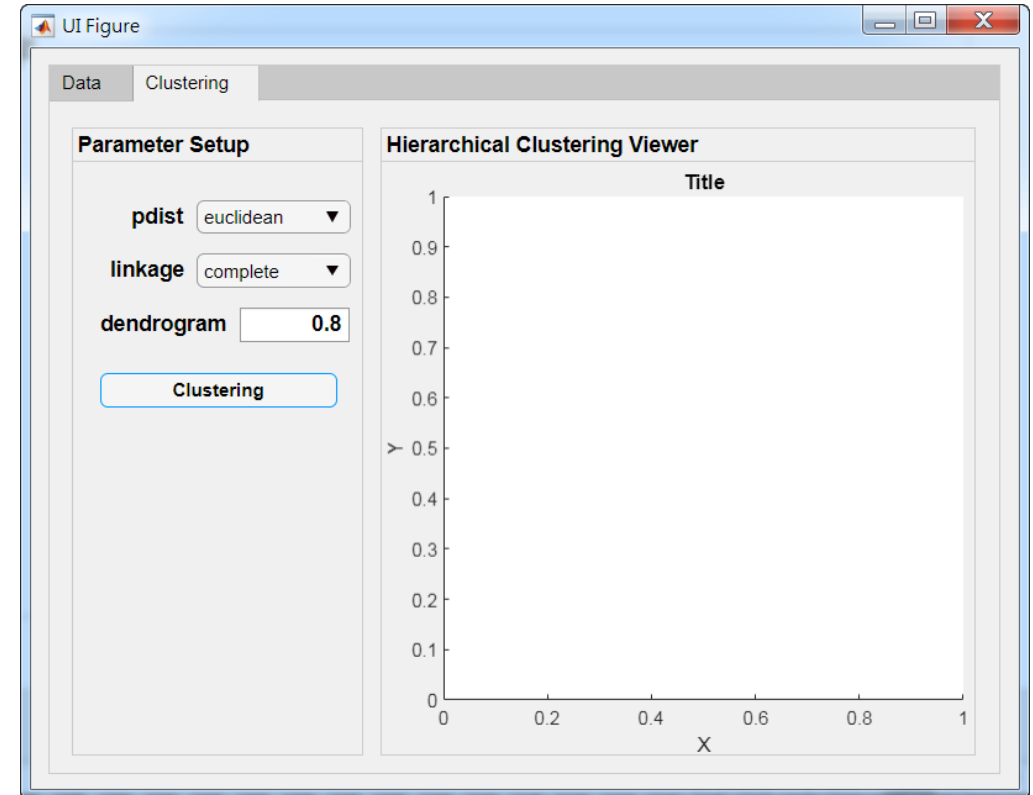
- CV & Publications
- Members
- Research Interests
- Teaching Materials**
- Download Platforms
- Activities
- Relevant Links

- MRI (UG)
- MRM (UG)
- MRI Research (G)
- MATLAB programming (UG)
- MATLAB ML (G)**
- MATLAB GUI (G)
- Signal Processing (G)
- Computer Sci. (UG)
- Computer Arch. (UG)
- fMRI Analysis (G)
- rs-fMRI Analysis (G)
- fNIRS Basics (G)
- fNIRS Workshop (G)
- Human Dissection (UG)
- Neuroanatomy (UG)
- Image Processing (R)

  **NIE**

Team Project Proposal

- One to two A4 Pages
- Contents
 - Project title
 - Team members (names and student IDs)
 - Brief background and goals
 - Data (resources, types, and numbers)
 - Data preprocessing
 - Design of GUI (using appdesigner), including layouts and function descriptions
- Email to alvin4016@nycu.edu.tw **before 5/2**
 - Please name the email title as “Matlab project: Group #”



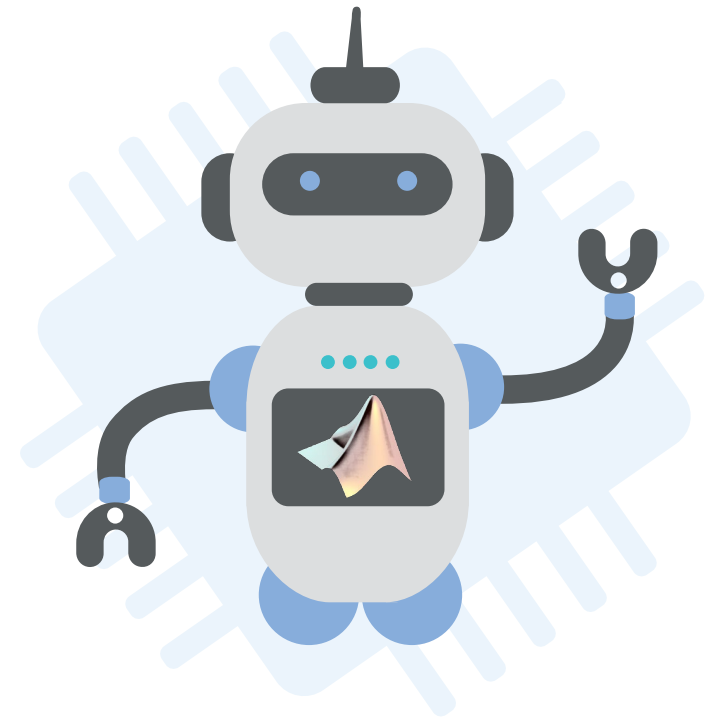
Contents in this Week

01 Introduction of App Designer

Build your very first App!

02 Advanced Usages of App Designer

Property and package of App





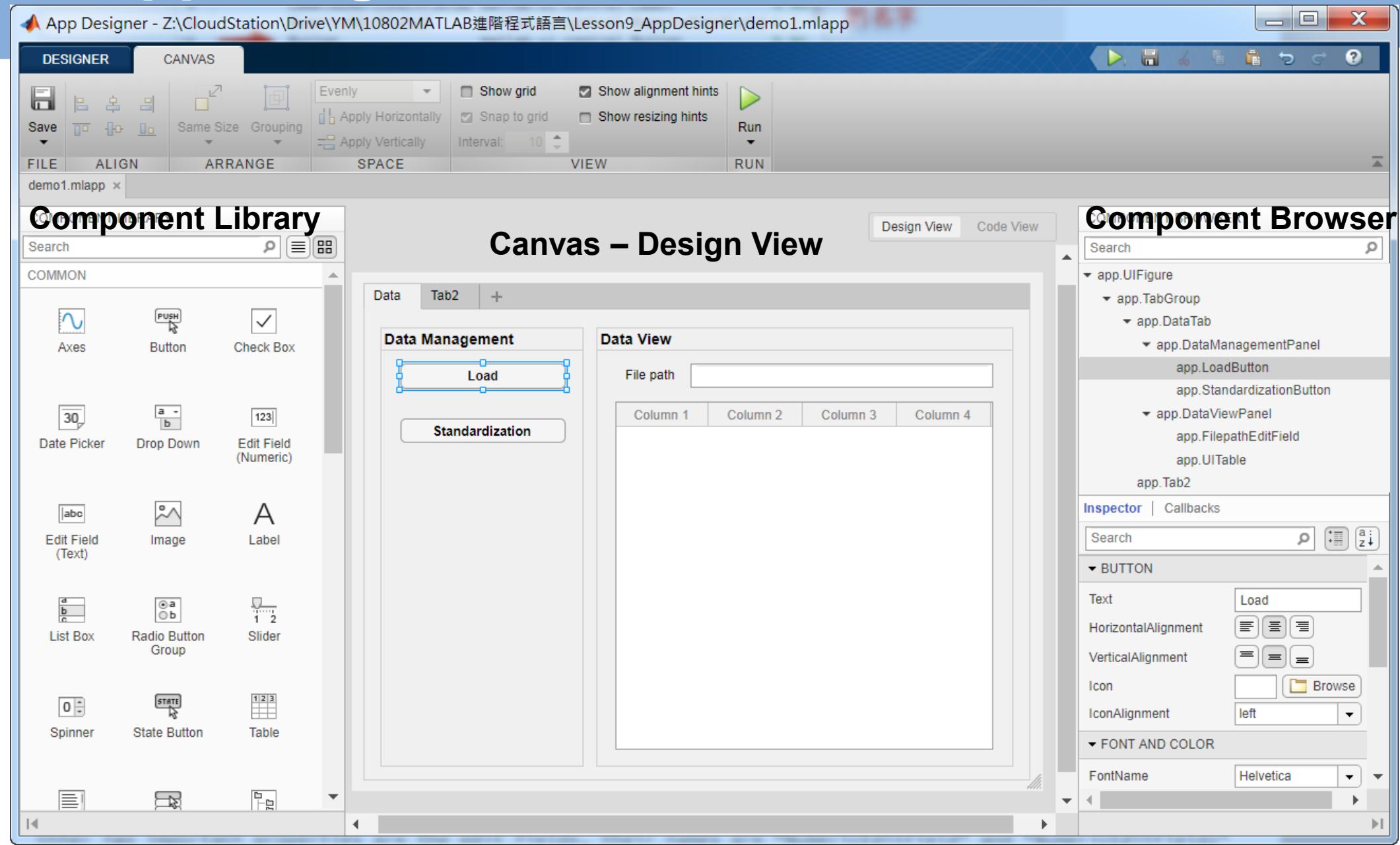
Introduction of App Designer

Build your very first App!

App Designer
















- Create professional apps without having to be a professional software developer.
- Design a User Interface
 - Drag and drop visual components to lay out the design of your graphical user interface (GUI).
- Define App Behavior
 - Use the integrated editor to quickly program its behavior.
- Package Your App




>> appdesigner



Component Library

COMMON

 Axes	 Button	 Check Box
 Date Picker	 Drop Down	 Edit Field (Numeric)
 Edit Field (Text)	 Image	 Label
 List Box	 Radio Button Group	 Slider
 Spinner	 State Button	 Table

 Text Area	 Toggle Button Group	 Tree
--	--	---

CONTAINERS















 Grid Layout	 Panel	 Tab Group
--	--	--

FIGURE TOOLS

 Menu Bar

INSTRUMENTATION

 90 Degree Gauge	 Discrete Knob	 Gauge
 Knob	 Lamp	 Linear Gauge
 Rocker Switch	 Semicircular Gauge	 Switch
 Toggle Switch		

>> appdesigner

The screenshot displays the MATLAB App Designer interface for a file named `demo1.mlapp`. The interface is divided into several panels:

- DESIGNER / EDITOR Tabs:** The **EDITOR** tab is active, showing the code editor.
- Toolbar:** Includes icons for Save, Callback, Function, Property, App Input Arguments, Go To, Find, Comment, Indent, Enable app coding alerts, Show Tips, and Run.
- Component Library:** Located on the left, it provides a search bar and instructions to add callback functions. Below it is the **APP LAYOUT** section, which shows a visual representation of the app's UI components.
- Canvas – Code View:** The central area displays the MATLAB code for the app. The code defines a class `demo1` that inherits from `matlab.apps.AppBase`. It includes properties for various UI components, a `createComponents` function to initialize them, and a `load` button callback.
- Component Browser:** Located on the right, it lists available components such as `app.UIFigure`, `app.TabGroup`, `app.DataTab`, `app.DataManagementPanel`, `app.LoadButton`, `app.StandardizationButton`, `app.DataViewPanel`, `app.FilepathEditField`, `app.UITable`, and `app.Tab2`.
- Inspector:** Below the Component Browser, it shows the properties of the selected component (a `BUTTON`). The `Text` property is set to `Load`, and the `FontName` is set to `Helvetica`.

```
classdef demo1 < matlab.apps.AppBase
% Properties that correspond to app components
properties (Access = public)
    UIFigure          matlab.ui.Figure
    TabGroup           matlab.ui.container.TabGroup
    DataTab            matlab.ui.container.Tab
    DataManagementPanel matlab.ui.container.Panel
    LoadButton        matlab.ui.control.Button
    StandardizationButton matlab.ui.control.Button
    DataViewPanel      matlab.ui.container.Panel
    FilepathEditFieldLabel matlab.ui.control.Label
    FilepathEditField  matlab.ui.control.EditField
    UITable            matlab.ui.control.Table
    Tab2               matlab.ui.container.Tab
end

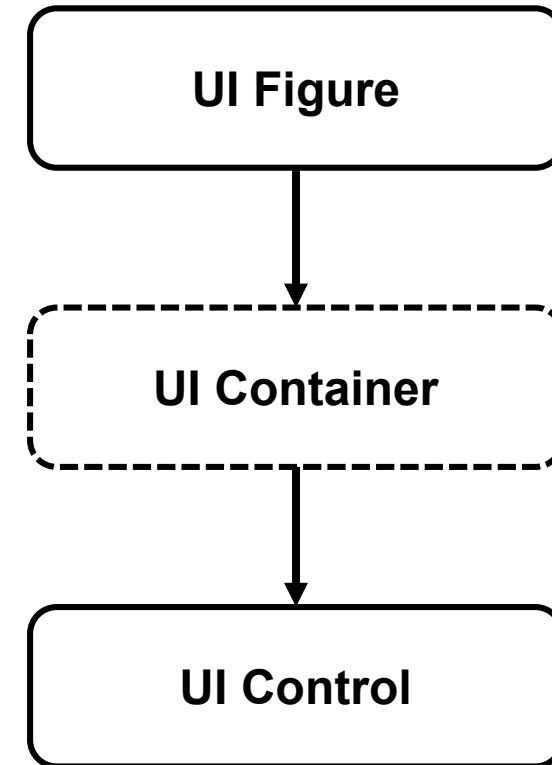
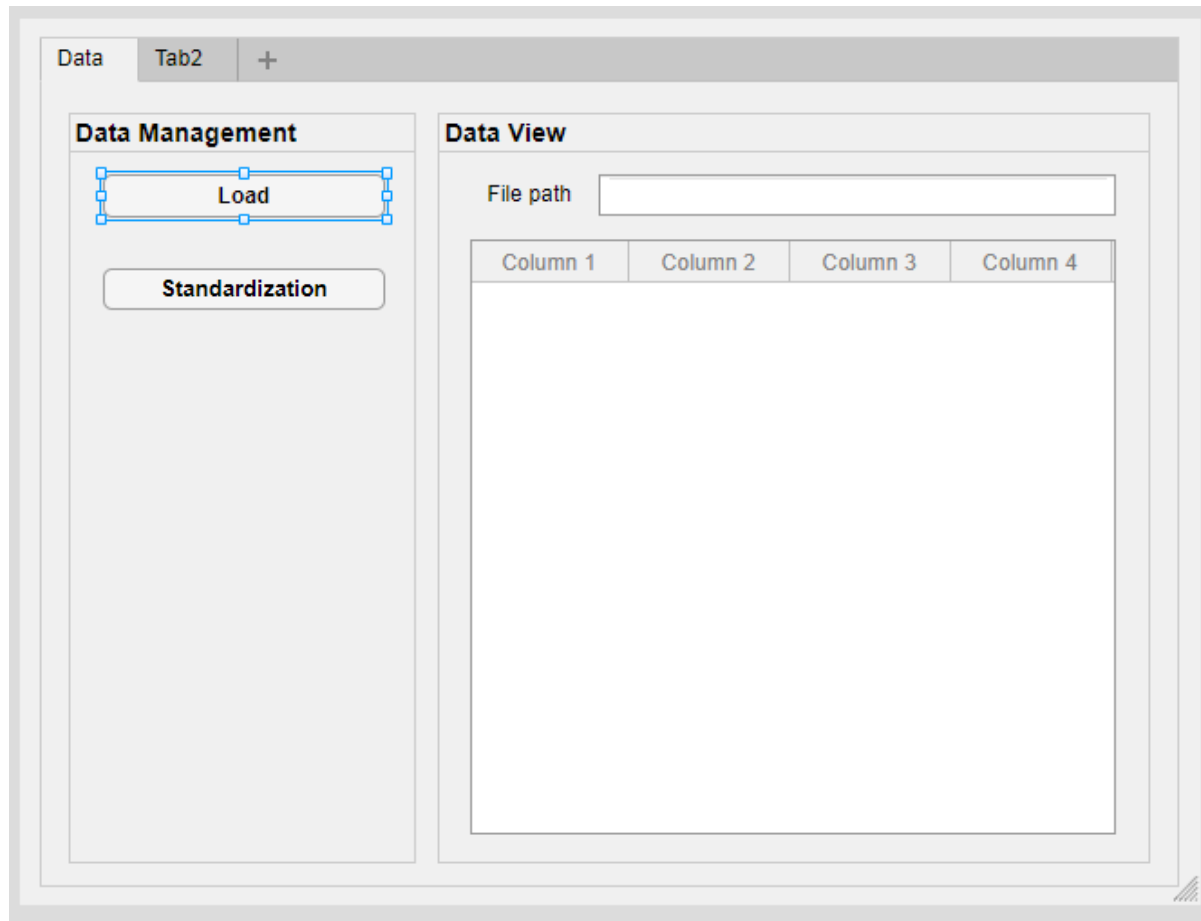
% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

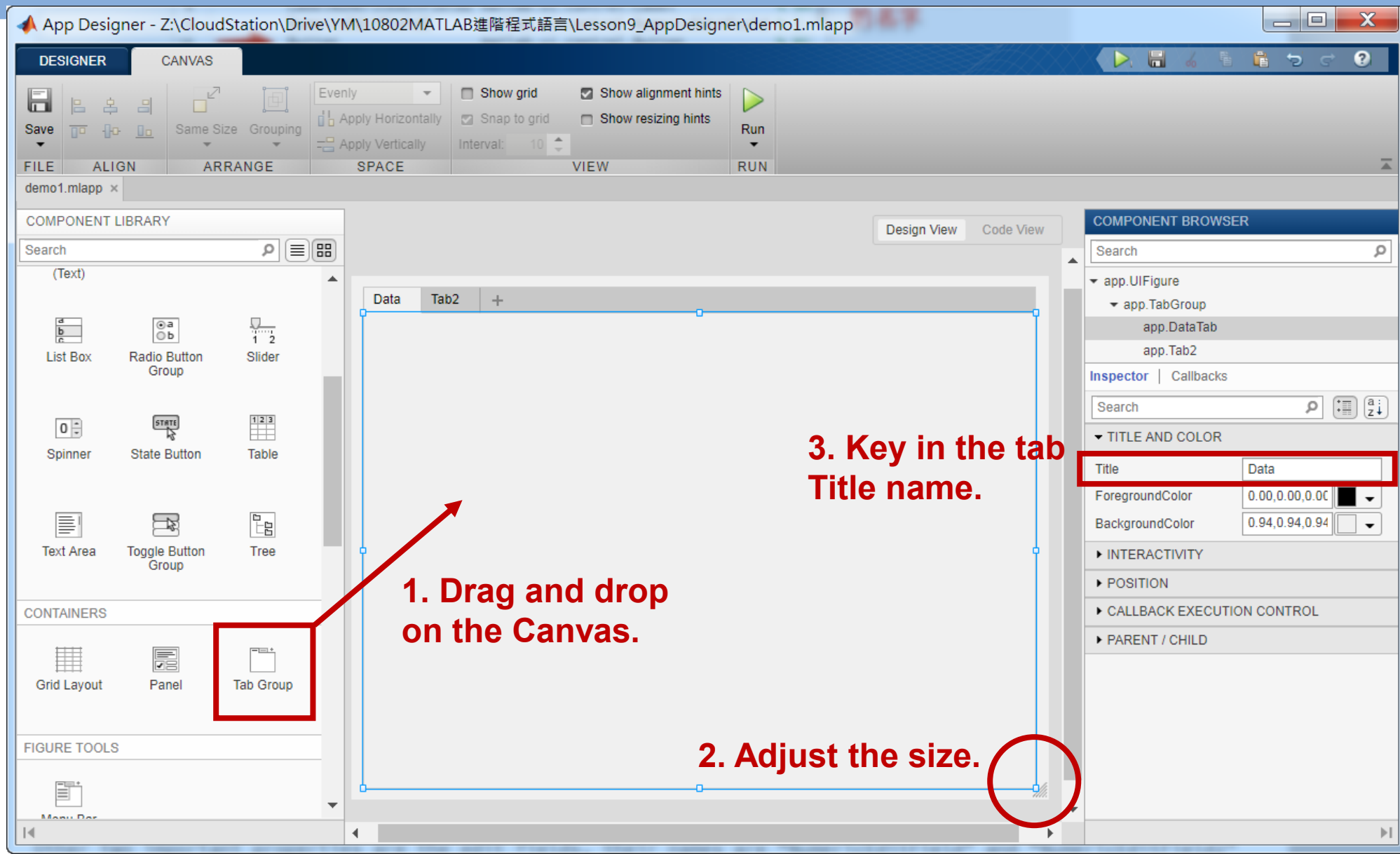
% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 640 480];
app.UIFigure.Name = 'UI Figure';

% Create TabGroup
```

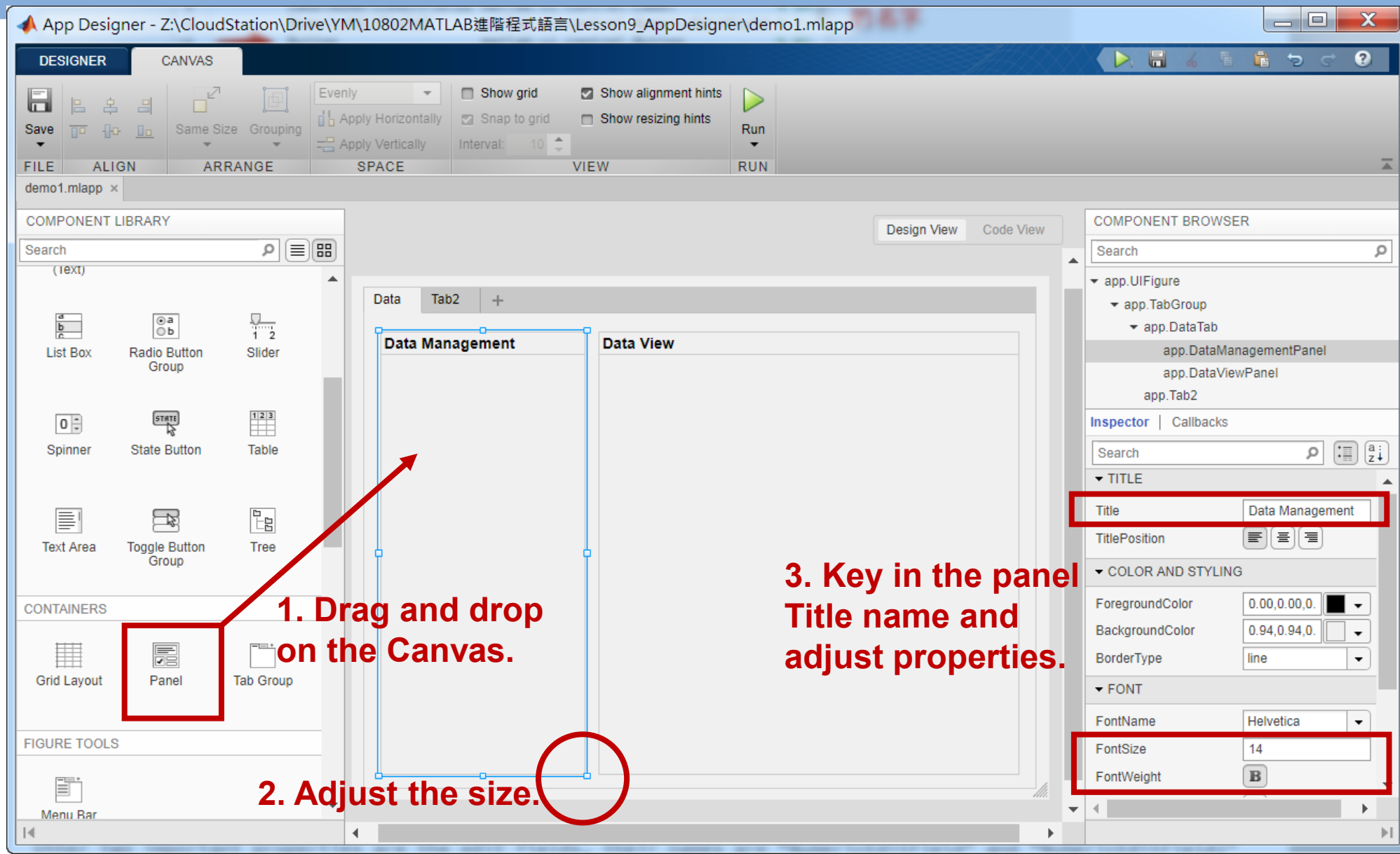
App Components



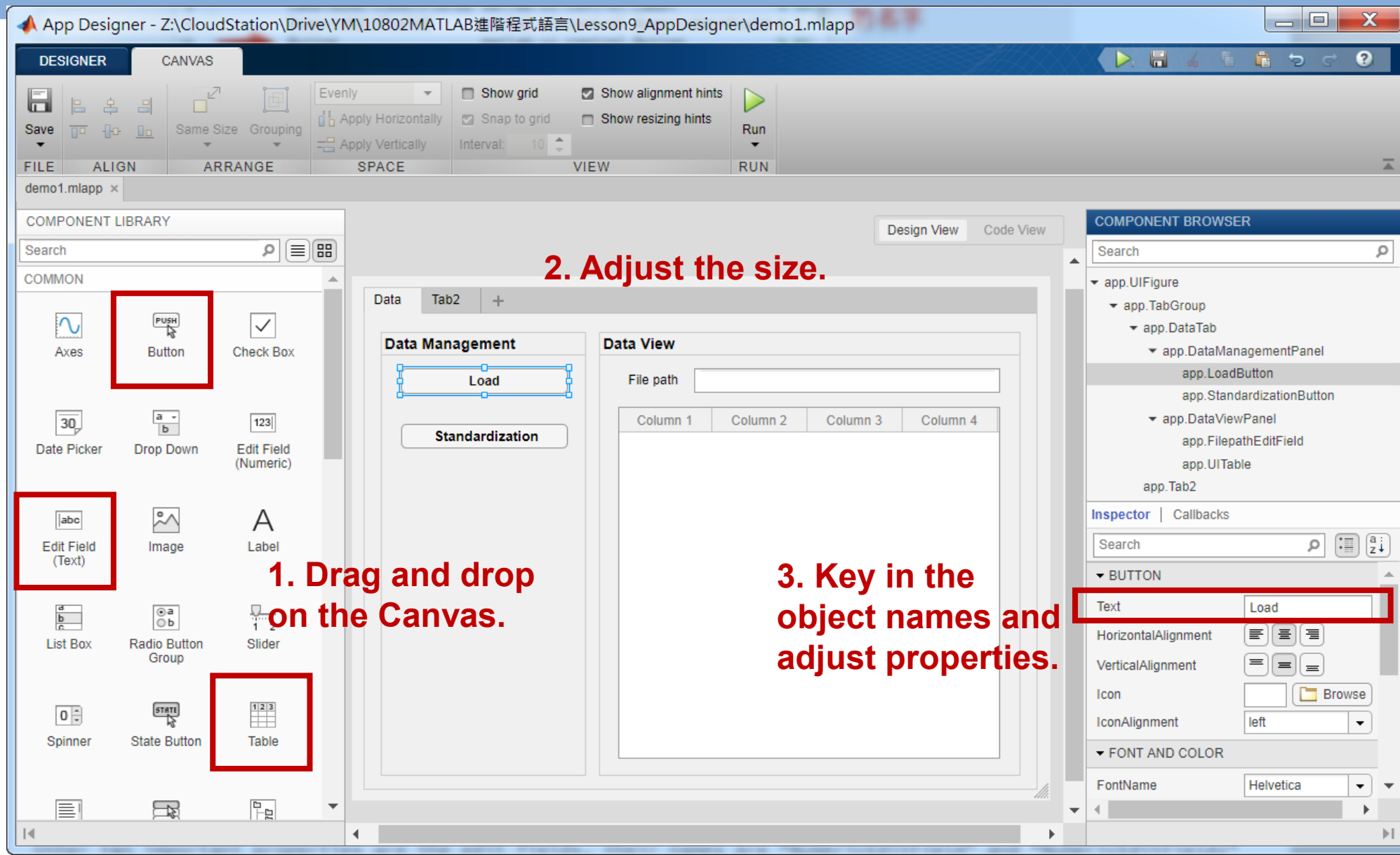
Create Your First App – UI Container



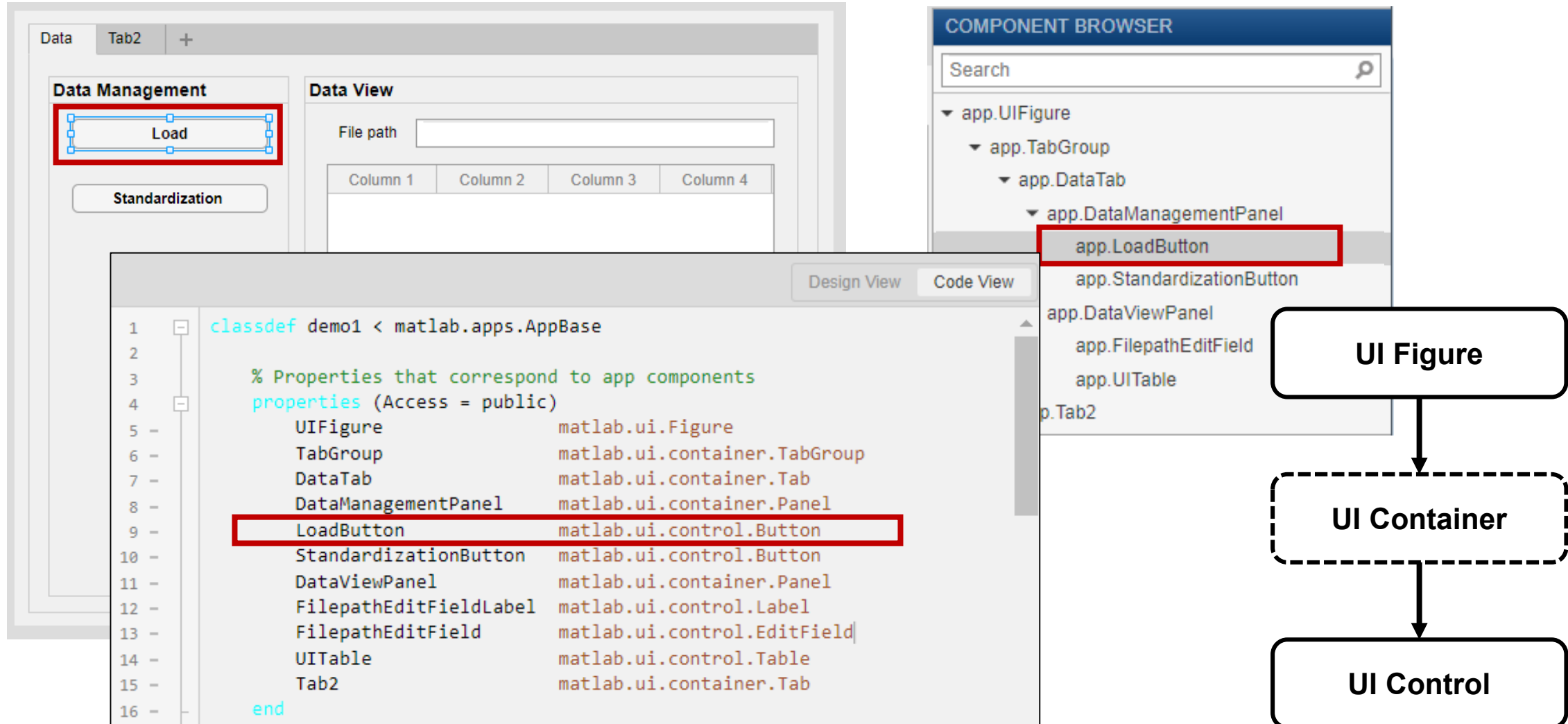
Create Your First App – UI Container



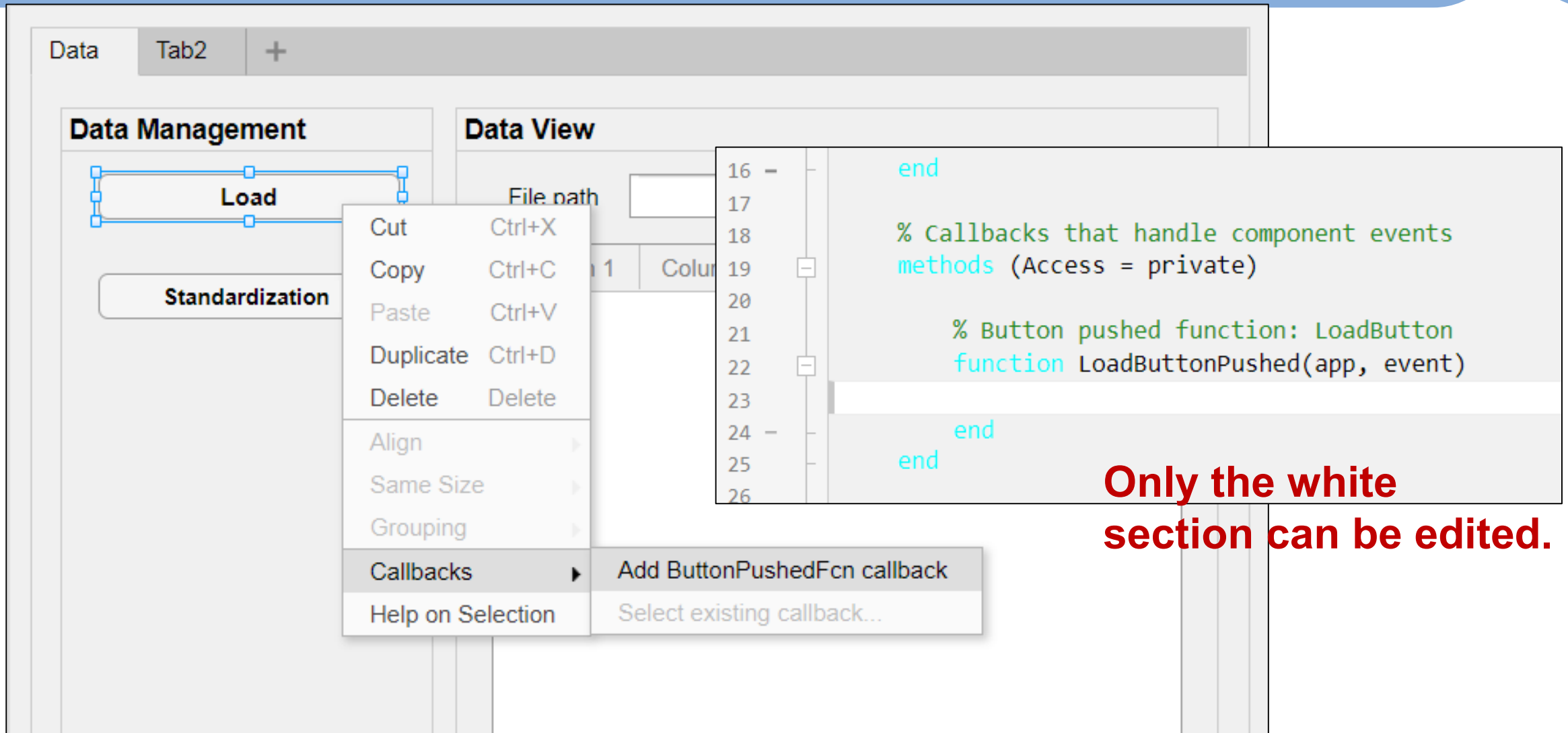
Create Your First App – UI Control



App Components



Callback of UI controls – Load



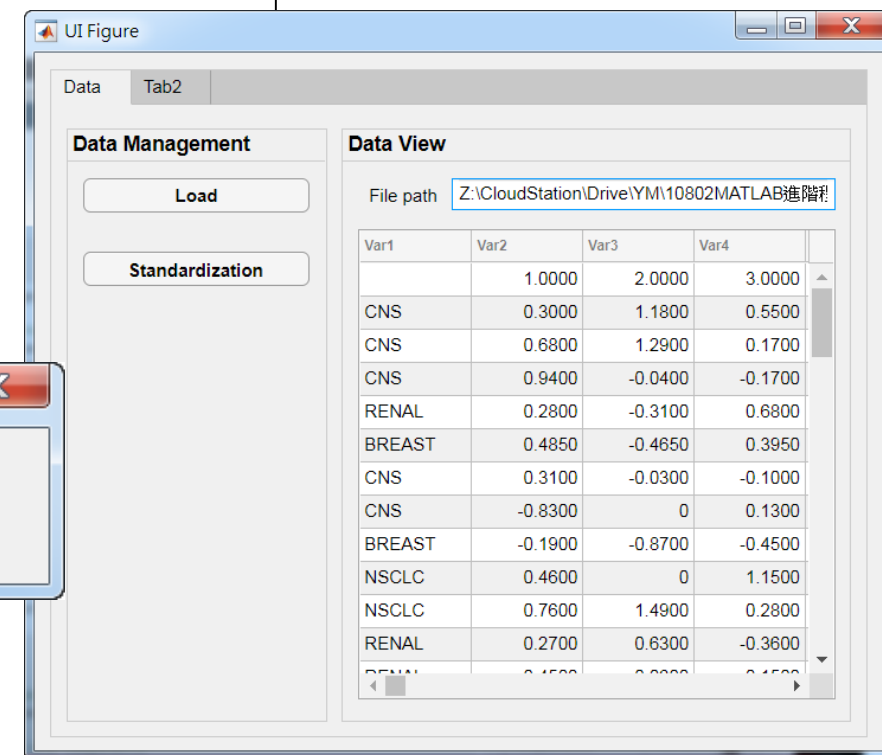
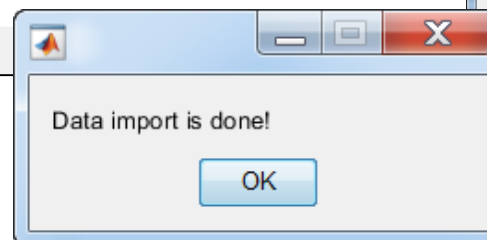
The screenshot displays the MATLAB App Designer interface. In the 'Data Management' section, a 'Load' button is selected, and a context menu is open. The 'Callbacks' option is highlighted, showing two choices: 'Add ButtonPushedFcn callback' and 'Select existing callback...'. The 'Data View' pane on the right shows the MATLAB code for the callback function, which is a function named 'LoadButtonPushed' that takes 'app' and 'event' as inputs. The code is as follows:

```
16 - end
17
18 % Callbacks that handle component events
19 methods (Access = private)
20
21 % Button pushed function: LoadButton
22 function LoadButtonPushed(app, event)
23
24 end
25 end
26
```

Only the white section can be edited.

Load data and display in UITable

```
18 % Callbacks that handle component events
19 methods (Access = private)
20
21 % Button pushed function: LoadButton
22 function LoadButtonPushed(app, event)
23     [filename,filepath] = uigetfile('*.csv','Please select a file to import. ');
24     if filename
25         app.FilepathEditField.Value = [filepath, filename];
26         data = readtable([filepath,filename]);
27         app.UITable.Data = data;
28         app.UITable.ColumnName = data.Properties.VariableNames;
29         msgbox('Data import is done!')
30
31         assignin('base','app',app) % assign 'app' to workspace
32     end
33 end
```



Load MLmaterials_L9\NCI60data.csv

Codes for Copy

```
[filename,filepath] = uigetfile('*.','Please select a file to import.');
```

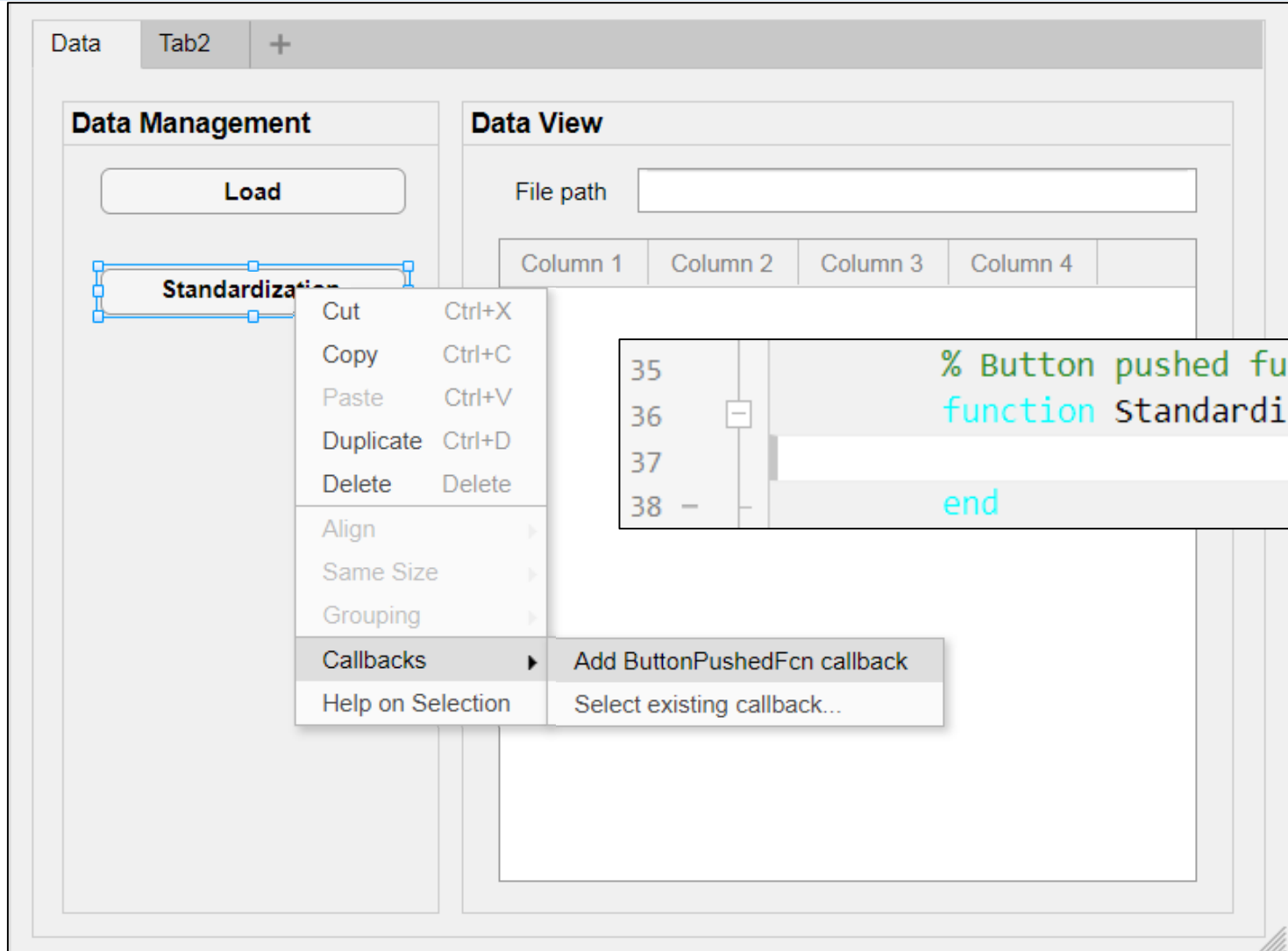
if filename

```
    app.FilepathEditField.Value = [filepath, filename];  
    data = readtable([filepath,filename]);  
    app.UITable.Data = data;  
    app.UITable.ColumnName = data.Properties.VariableNames;  
    msgbox('Data import is done!')
```

assignin('base','app',app) % assign 'app' to workspace

```
end
```

Callback of UI controls – Standardization



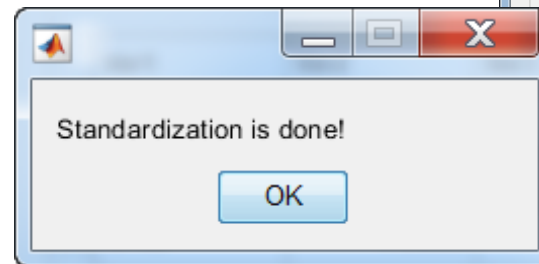
The screenshot shows the MATLAB App Designer interface. On the left, the 'Data Management' pane contains a 'Load' button and a 'Standardization' button. The 'Standardization' button is selected, and a context menu is open over it. The menu includes options like 'Cut', 'Copy', 'Paste', 'Duplicate', 'Delete', 'Align', 'Same Size', 'Grouping', 'Callbacks', and 'Help on Selection'. The 'Callbacks' option is expanded, showing 'Add ButtonPushedFcn callback' and 'Select existing callback...'. On the right, the 'Data View' pane shows a table with four columns: 'Column 1', 'Column 2', 'Column 3', and 'Column 4'. Below the table, a code editor displays the following MATLAB code:

```
% Button pushed function: StandardizationButton
function StandardizationButtonPushed(app, event)

end
```

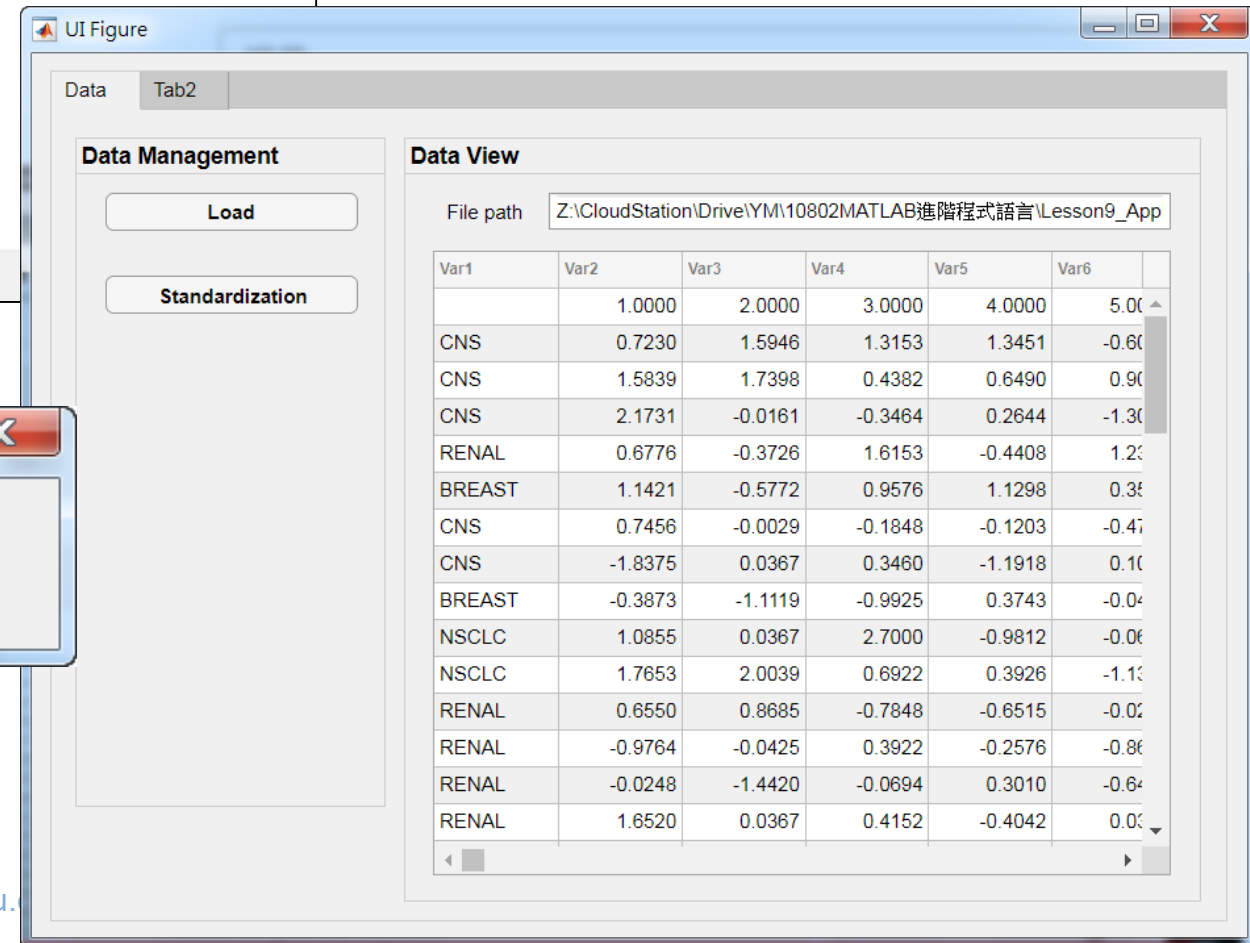
Standardize the data

```
35 % Button pushed function: StandardizationButton
36 function StandardizationButtonPushed(app, event)
37     data = app.UITable.Data;
38     numdata = table2array(data(2:end,2:end));
39     Z = zscore(numdata);
40     data(2:end,2:end) = array2table(Z);
41     app.UITable.Data = data;
42
43     msgbox('Standardization is done!')
44 end
```



MLmaterials_L9\demo1.mlapp

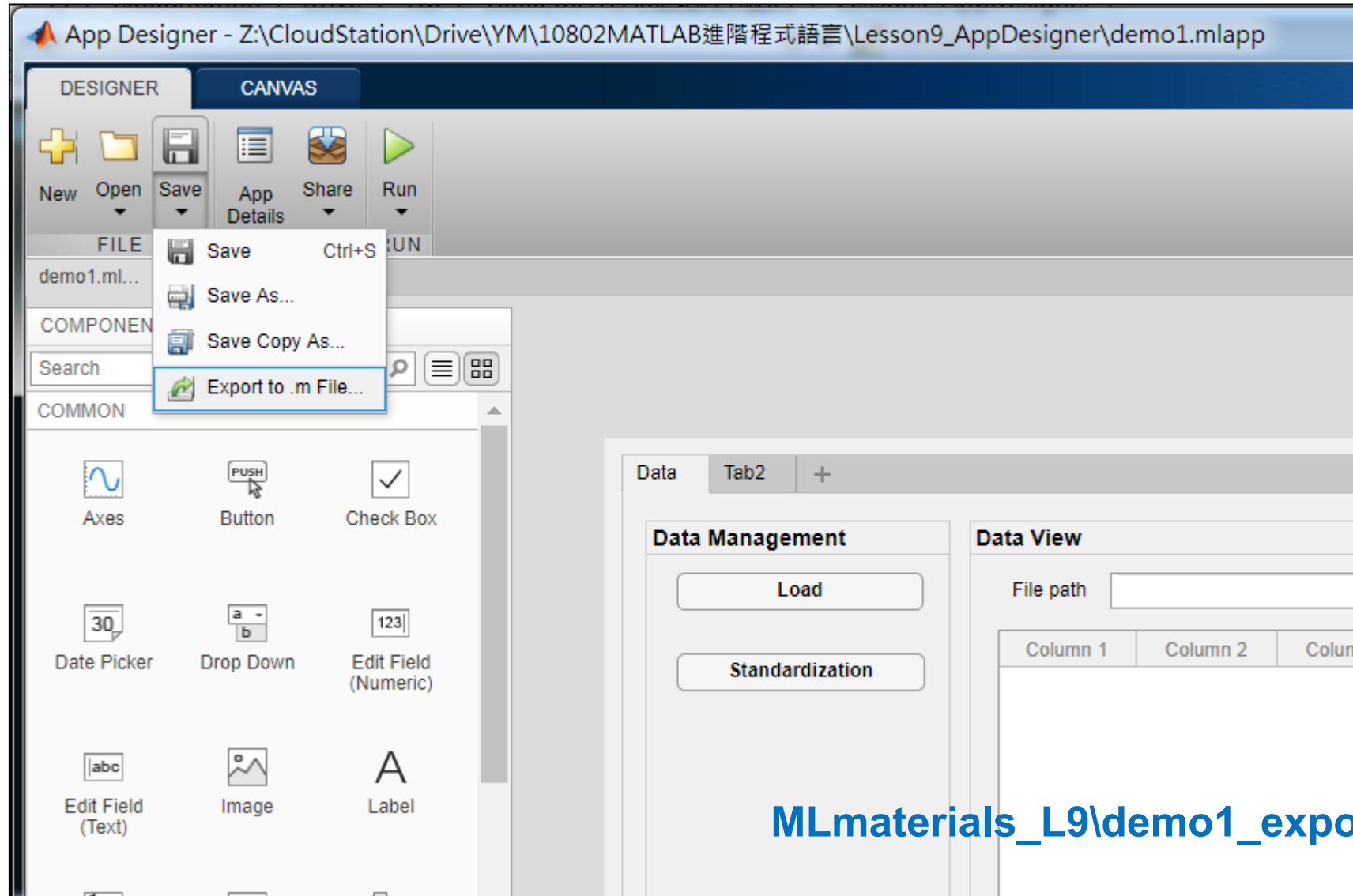
<http://cflu.lab.nycu.edu.tw/>




Codes for Copy

```
data = app.UITable.Data;  
numdata = table2array(data(2:end,2:end));  
Z = zscore(numdata);  
data(2:end,2:end) = array2table(Z);  
app.UITable.Data = data;  
  
msgbox('Standardization is done!')
```

Generate M-code (*.m)



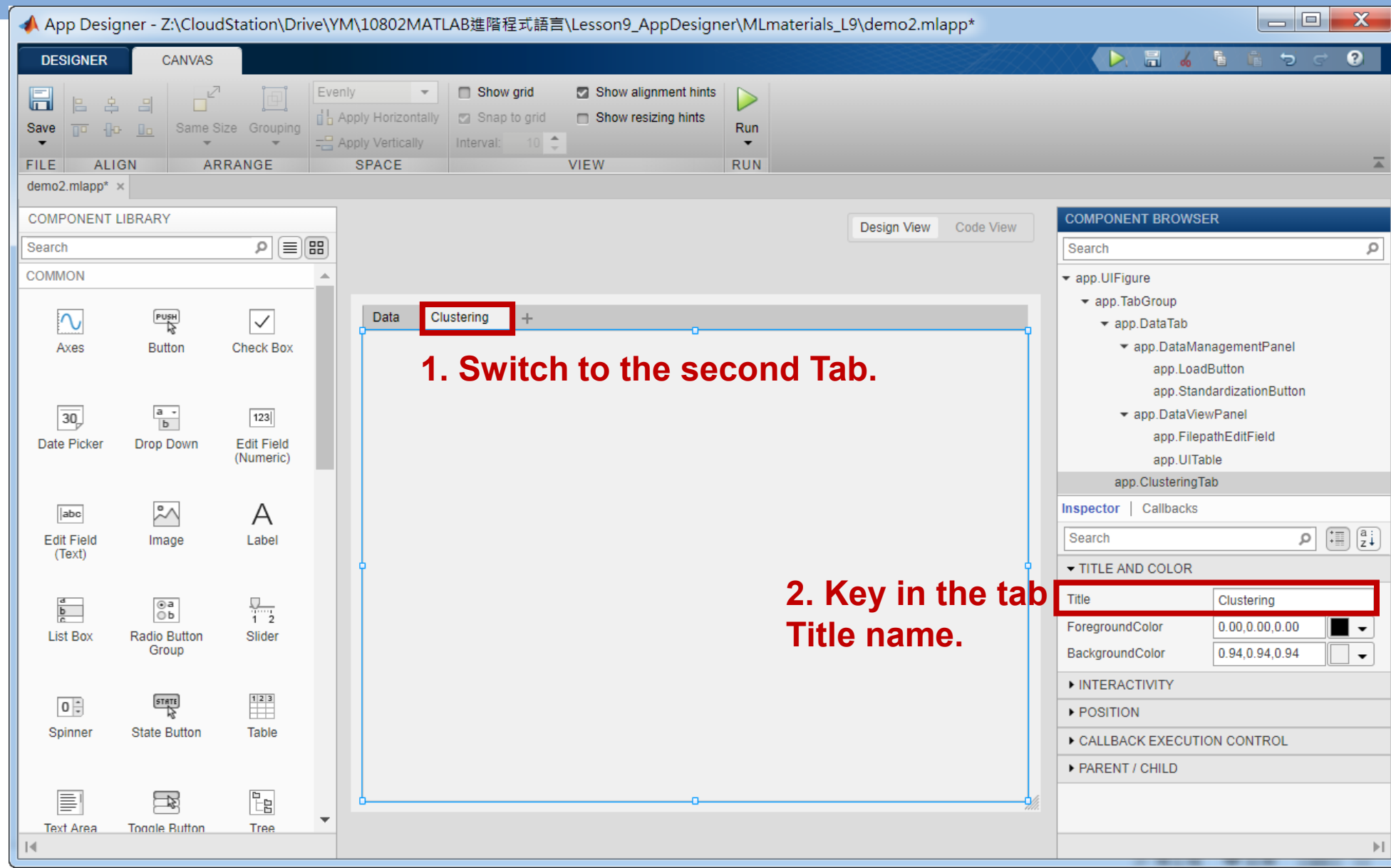
MLmaterials_L9\demo1_exported.m



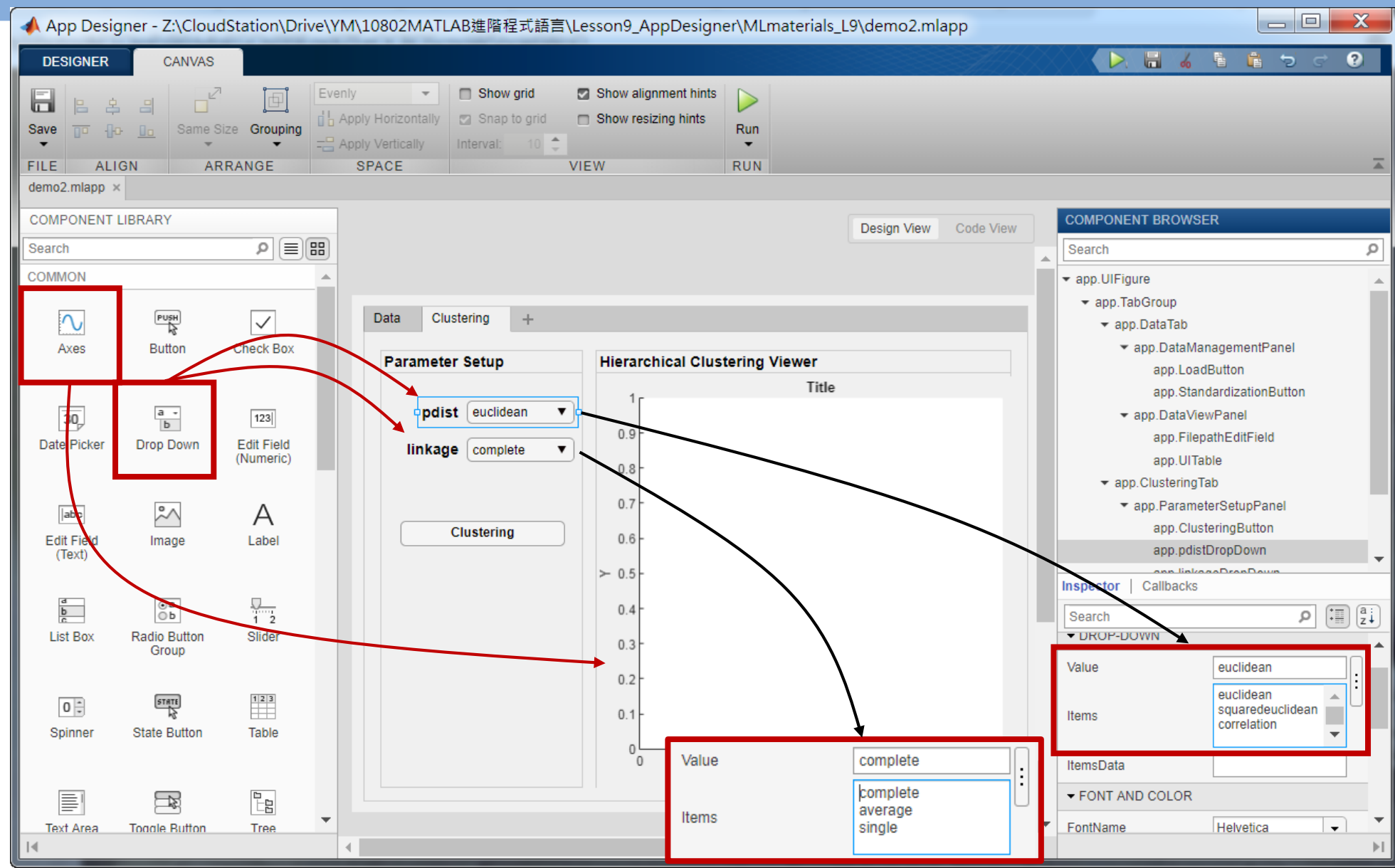
Advanced Usages of App Designer

Property and package of App

UI Tab Usage



Layout of the second Tab



Perform Hierarchical Clustering

```
70 % perform hierarchical clustering
71 - data = app.UITable.Data;
72 - numdata = table2array(data(2:end,2:end));
73 - datalabel = table2cell(data(2:end,1));
74
75 - D=pdist(numdata,app.pdistDropDown.Value);
76 - Ctree=linkage(D,app.linkageDropDown.Value);
77
78 - [H,T,outperm]=dendrogram(Ctree,0,'Labels',datalabel,'Orientation','top');
79
80 - plot(app.UIAxes,0)
81 - hold(app.UIAxes) % hold axes
82 - for i=1:length(H)
83 -     plot(app.UIAxes,H(i).XData,H(i).YData,'color','b')
84 - end
85 - hold(app.UIAxes) % release axes
86 - close(gcf) % close the figure generated by dendrogram
87
88 - app.UIAxes.XTick = 1:length(datalabel);
89 - app.UIAxes.XTickLabel = datalabel(outperm);
90 - app.UIAxes.XTickLabelRotation = 90;
91 - app.UIAxes.XLabel.String = '';
92 - app.UIAxes.YLabel.String = '';
93 - app.UIAxes.YGrid = 'on';
```

MLmaterials_L9\demo2.mlapp

Codes for Copy

```
data = app.UITable.Data;
numdata = table2array(data(2:end,2:end));
datalabel = table2cell(data(2:end,1));
D=pdist(numdata,app.pdistDropDown.Value);
Ctree=linkage(D,app.linkageDropDown.Value);
[H,T,outperm]=dendrogram(Ctree,0,'Labels',datalabel,'Orientation','top');
plot(app.UIAxes,0)
hold(app.UIAxes) % hold axes
for i=1:length(H)
    plot(app.UIAxes,H(i).XData,H(i).YData,'color','b')
end
hold(app.UIAxes) % release axes
close(gcf) % close the figure generated by dendrogram
app.UIAxes.XTick = 1:length(datalabel);
app.UIAxes.XTickLabel = datalabel(outperm);
app.UIAxes.XTickLabelRotation = 90;
app.UIAxes.XLabel.String = "";
app.UIAxes.YLabel.String = "";
app.UIAxes.YGrid = 'on';
```

Add in a Property under app

The screenshot displays the MATLAB App Designer interface. The top toolbar includes buttons for Save, Callback, Function, Property, and App Input Arguments. The 'Property' button is highlighted with a red box. Below the toolbar, the 'CODE BROWSER' panel is visible, showing a list of properties. The 'Properties' tab is selected, and a search bar is present. A tooltip is displayed over the 'Property' button, explaining the difference between Private and Public properties.

Private Property
Private properties store data to be shared within the app only

Public Property
Public properties store data to be shared inside and outside of the app

CODE BROWSER
Callbacks | Functions | **Properties**

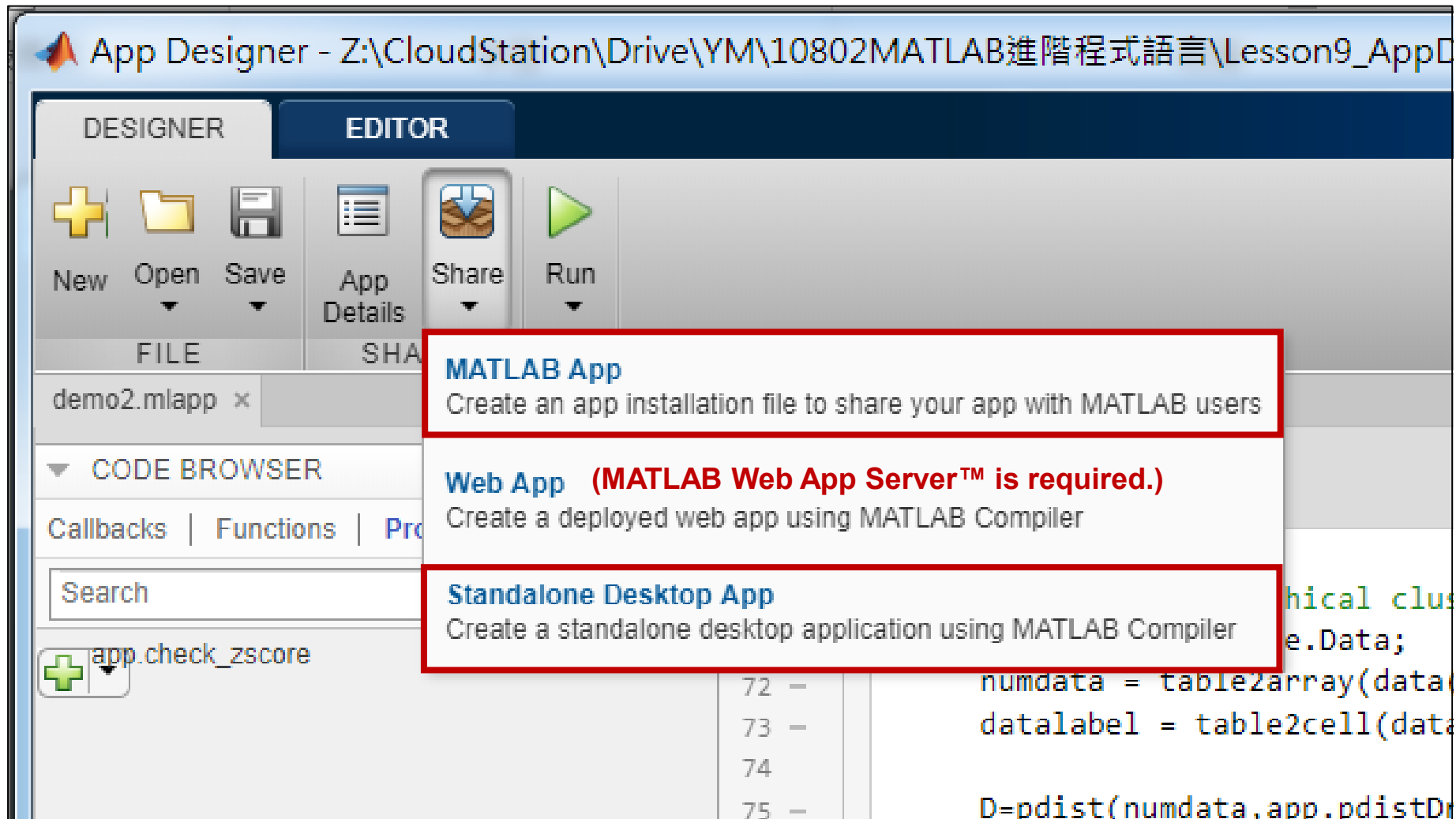
Search

Add a property to create a variable to store and share data between callbacks and functions. Specify the property name with the prefix `app.` to access the property value:

```
app.Property = someData;
```

19	-	linkageDropDownLabel	matl
20	-	linkageDropDown	matl
21	-	dendrogramEditFieldLabel	matl
22	-	dendrogramEditField	matl
23	-	pdistDropDown	matl
24	-	HierarchicalClusteringViewerPa	

Standalone Desktop App





THE END

Contact:

盧家鋒 alvin4016@nycu.edu.tw