


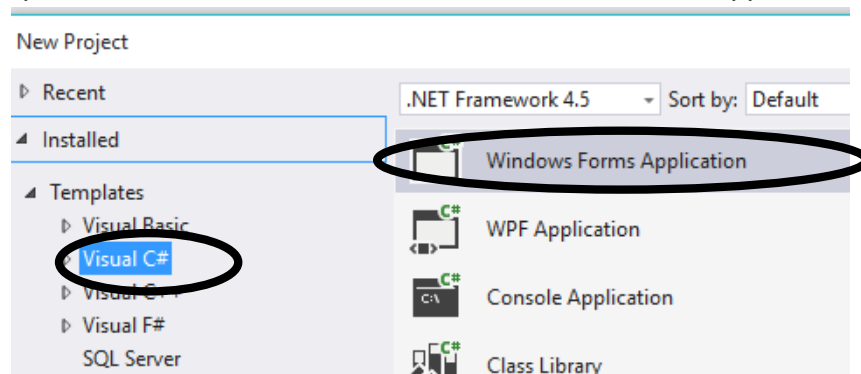


Case 5 A Simple Standalone GUI for R Apps

You are to create a C# application that takes in any CSV file and summarize it based on the selected columns by the user. The data processing will be done by R in the backend.

1. Creating a C# Application

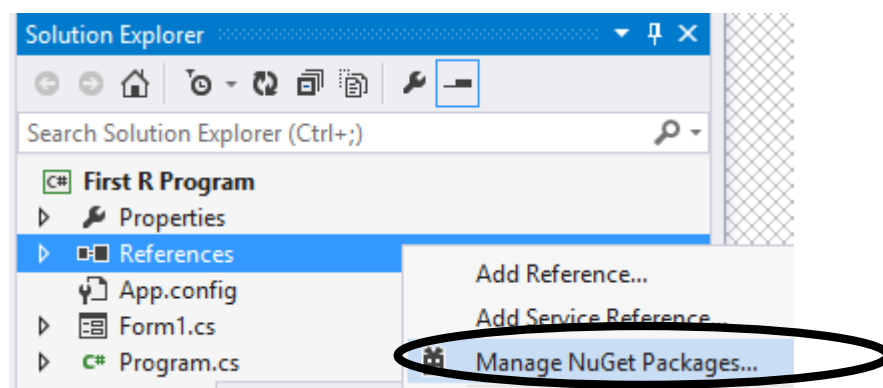
- 1.1. Open Visual Studio 2013 
- 1.2. Click on File-> New Project
- 1.3. In the Templates Tab, Choose Visual C# then Windows Forms Application



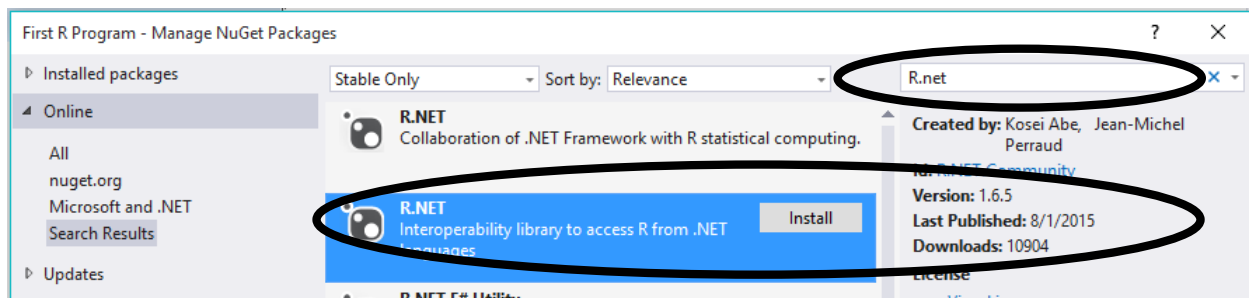
- 1.4. Enter a Name for the project as: "First R Program"
- 1.5. Click on Ok.

2. Adding References


- 2.1. If this is your first time to create a R+C# program, we need to add packages that enable the communication of C# and R. Click on Tools->Extensions and Updates
- 2.2. In the Search Text Field, Type NuGet Package Manager.
- 2.3. Click on Install.
- 2.4. Click on Close when done. Steps 2.1 to 2.3 need to be done only once during the first implementation.
- 2.5. In the solution explorer right click on References and click on Manage NuGet Packages



- 2.6. In the Manage NuGet Packages Window, type "R.NET" in the search Bar.



2.7. Click on Install. Make sure that Version 1.6.5 is to be installed.

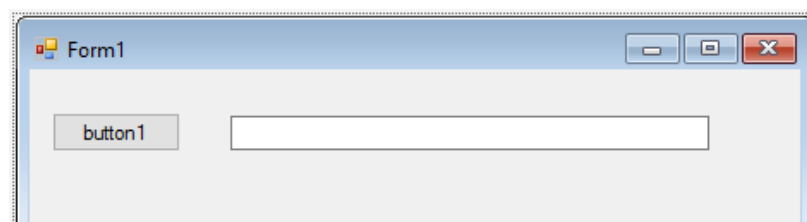
2.8. If the check sign is enabled , click on Close.

3. Designing the Application GUI

3.1. From the toolbox tab, Click on the “Button” operator and drag it into the Form1 window.

3.2. From the toolbox tab, Click on the “TextBox” operator and drag it into the Form1 window.

3.3. The Layout of Form1 should look like this:



3.4. Click on button1 then in the properties panel, change the following:

Property	Value
(name)	btnSearch
Text	Search

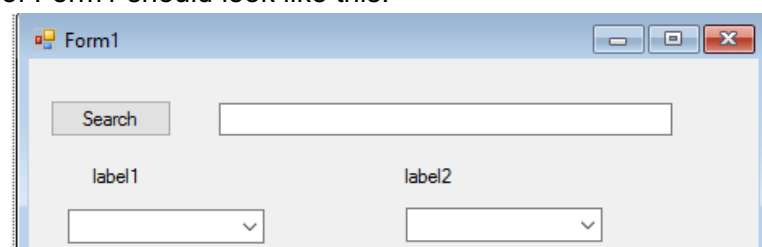
3.5. Click on the textbox1 then in the properties panel, change the following:

Property	Value
(name)	txtSearch

3.6. From the toolbox tab, Click on the “Label” operator and drag it into the Form1 window. Drag another “Label” operator into the Form1 window.

3.7. From the toolbox tab, Click on the “ComboBox” operator and drag it into the Form1 window. Drag another “ComboBox” operator into the Form1 window.

3.8. The Layout of Form1 should look like this:



3.9. Click on each label then in the properties panel, change the following:

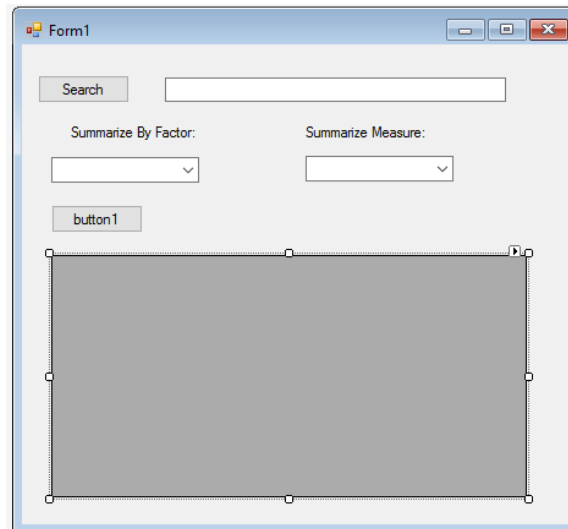
Property	Value	Property	Value
(name)	lbl1	(name)	lbl2



- 3.10. Click on each combo box then in the properties panel, change the following:

Property	Value	Property	Value
(name)	cmbby	(name)	cmbmeas

- 3.11. From the toolbox tab, Click on the “Button” operator and drag it into the Form1 window.
- 3.12. From the toolbox tab, Click on the “DataGridView” operator and drag it into the Form1 window. The layout of Form1 should look like this:



- 3.13. Click on button1 then in the properties panel, change the following:

Property	Value
(name)	btnAnalyze
Text	Analyze

- 3.14. From the toolbars tab, drag an openFileDialog box onto the layout.

4. C# Coding

- 4.1. Double click on the Search Button

- 4.2. Right after the `using System.Windows.Forms;` line type the following:

- `using RDotNet;`

- 4.3. Inside the `btnSearch_Click()` function, type the following lines of code:

- `if (openFileDialog1.ShowDialog() == DialogResult.OK)`
- `{`
- `string test = openFileDialog1.FileName;`
- `string test2= test.Replace("\\", "/");`
- `txtSearch.Text = test2;`
- `}`
- `REngine engine = REngine.GetInstance();`
- `string Command = "dataset<-read.csv(file='" + txtSearch.Text +`
- `"',head=TRUE)";`
- `DataFrame dataset = engine.Evaluate(Command).AsDataFrame();`
- `for (int i = 0; i < dataset.ColumnCount; ++i)`
- `{`
- `cmbby.Items.Add(dataset.ColumnNames[i]);`

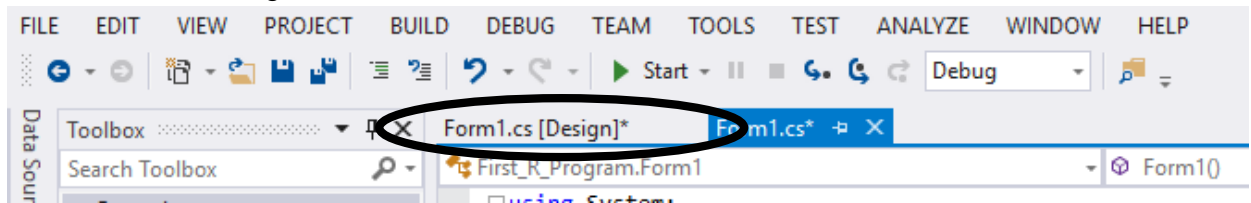
ERLJalao

Copyright UP National Engineering Center, eljalao@up.edu.ph



- `cmbmeas.Items.Add(dataset.ColumnNames[i]);`
- `}`

4.4. Go back to the Design tab:



4.5. Double click on the Analyze Button.

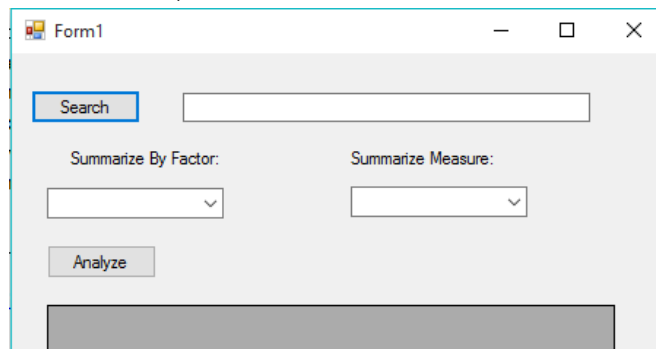
4.6. Inside the `btnAnalyze_Click()` function, type the following lines of code:

- `REngine engine = REngine.GetInstance();`
- `string Command = "dataset<-read.csv(file='"+ txtSearch.Text +`
`"',head=TRUE)";`
- `DataFrame dataset = engine.Evaluate(Command).AsDataFrame();`
- `engine.Evaluate("library('reshape2')");`
- `DataFrame datasetm = engine.Evaluate("dataset.m <- melt(dataset, id=c('" +`
`cmbby.SelectedItem + "')', measure=c('" + cmbmeas.SelectedItem +`
`"'))").AsDataFrame();`
- `DataFrame datasetc = engine.Evaluate("dataset.c = dcast(dataset.m, " +`
`cmbby.SelectedItem + " ~ variable, mean)").AsDataFrame();`
- `for (int i = 0; i < datasetc.ColumnCount; ++i)`
- `{`
- `dataGridView1.ColumnCount++;`
- `dataGridView1.Columns[i].Name = datasetc.ColumnNames[i];`
- `}`
- `for (int i = 0; i < datasetc.RowCount; ++i)`
- `{`
- `dataGridView1.RowCount++;`
- `dataGridView1.Rows[i].HeaderCell.Value = datasetc.RowNames[i];`
- `for (int k = 0; k < datasetc.ColumnCount; ++k)`
- `{`
- `dataGridView1[k, i].Value = datasetc[i, k];`
- `}`
- `}`
- `engine.Dispose();`
- `btnAnalyze.Enabled = false;`

5. Testing and Publishing the R/C# Program

5.1. Click on the Start button  to test the program.

5.2. If the following screen shows, click on the Search button



5.3. In the open file dialog window search for the Art SQL Extract and click on Open.

ERLJalao

Copyright UP National Engineering Center, eljalao@up.edu.ph



- 5.4. In the Summarize by Factor Dropdown, Select Condition while in the Summarize Measure Select Appraised Value

Summarize By Factor:	Summarize Measure:
Condition	Appraised_Value

- 5.5. Click on the Analyze button.

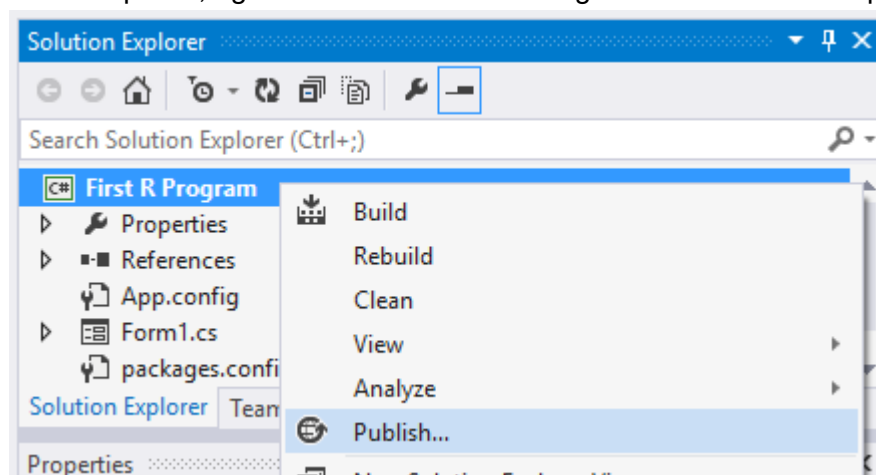
- 5.6. The following summarized result should be displayed:

	Condition	Appraised_Value
1	Excellent	11373.08235294...
2	Fair	7384.571428571...
3	Good	4698.277777777...
4	Poor	2530
*		

- 5.7. If the result is similar, then the program is ready for publication to create an Installer.

Close the program and go back to Visual Studio.

- 5.8. In the solution explorer, right click on the First R Program line and click on publish.



- 5.9. To specify a location to save the installer files, click on Browse.



- 5.10. In the desktop, create a new folder and name it First R Program and click on Open by selecting the new Folder.

- 5.11. Click on Next

- 5.12. In “How Will Users Install the Application” choose From a CD-ROM or DVD-ROM.

- 5.13. Click on Next

- 5.14. In “Where will the application check for updates?” choose the Application will not check for Updates.

- 5.15. Click on Next

- 5.16. Click on Finish.

6. Installing and Testing the Program



- 6.1. To test the standalone program, go to the desktop First R Program folder and Run Setup.
- 6.2. If the installation fails, it means that there exists a similar program named First R Program (from previous trainings). Just uninstall the program from the control panel then try installing.
- 6.3. Congratulations! You have created your first R/C# Standalone Program

The screenshot shows a Windows-style application window titled "Form1". Inside the window, there is a "Search" button followed by a text input field. Below these, there are two labels: "Summarize By Factor:" and "Summarize Measure:", each followed by a dropdown menu. Further down is an "Analyze" button. At the bottom of the window is a large, empty gray rectangular area, likely intended for displaying results or data.