Quickstart Guide to GeoVISTA *Studio* :

Installation

Before you can run *Studio*, you must install (or verify that you have) the latest version of Java, available at http://java.sun.com/ . Either the runtime environment (JRE) or software development kit (J2SDK) will work. If you are interested in development using Java, you may wish to opt for the SDK instead.

Once you have installed the latest version of Java, you can install the application. Begin by navigating to: http://www.geovistastudio.psu.edu/autobuild/

Click on the "Full" link to start installing GeoVISTA *Studio*.

During the installation process, you may be prompted with a warning message before the install proceeds. Please ignore this message and continue. *Studio* will load once it has finished downloading the components it needs.

You can create a desktop shortcut to *Studio* through the Java Web Start program by running javaws.exe and clicking on Application -> Create Shortcuts. Web Start is usually found in **C:\Program Files\Java Web Start** .

Using An Existing Design

A set of sample designs and this tutorial are available at:

http://www.geovistastudio.psu.edu/autobuild/

To load an existing design, click on "File"->"Load Design" and navigate to the directory where your designs are located. The default working directory can be set by clicking on "Preferences"->"User Preferences." Once you have selected a design file (.gvd) in the file navigator, click the "Open" button and the design will load.

The left panel is the *Studio* DesignBox. This is where applications are built using the *Studio* visual programming tools. The right panel is the *Studio* GUIBox, where the constructed applications and their GUI are displayed, thus where you can load and examine data.

To start using a design immediately, try loading any of the four designs included with the designs .zip file. Then, maximize the GUIBox window. Click on the "Select" button in the file chooser object (it should be at the top left of your screen) and navigate to the shapefile you'd like to examine. Select this file and click the "Open" button. The data will load and you can then explore with the visualization tools.

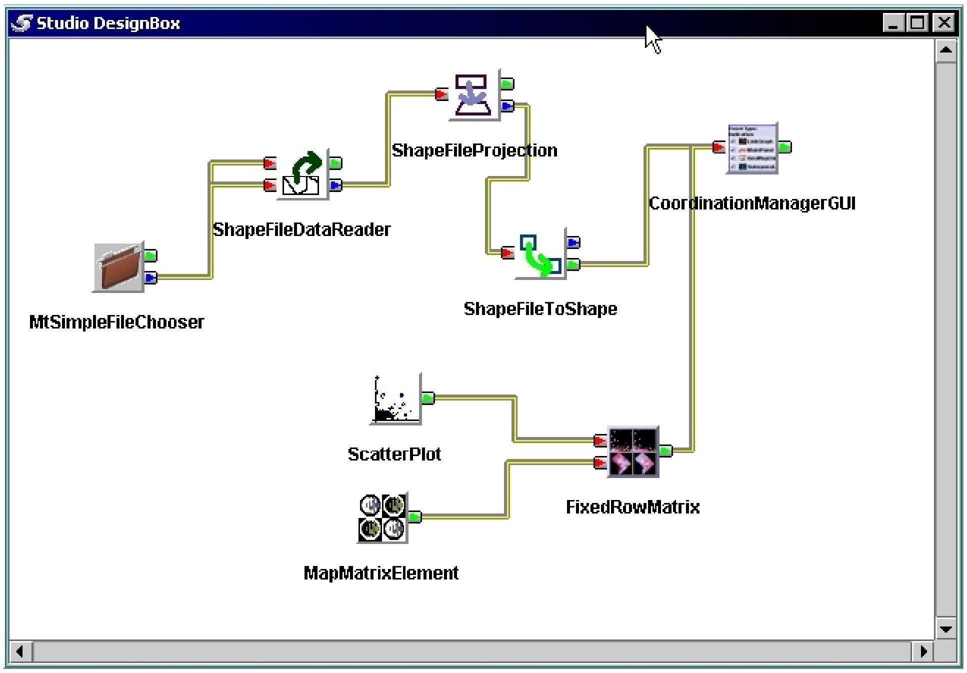
Creating Your Own Design

Studio installs with a large complement of application components called JavaBeans. These Beans can be linked together in the DesignBox to create applications for exploring and analyzing data.

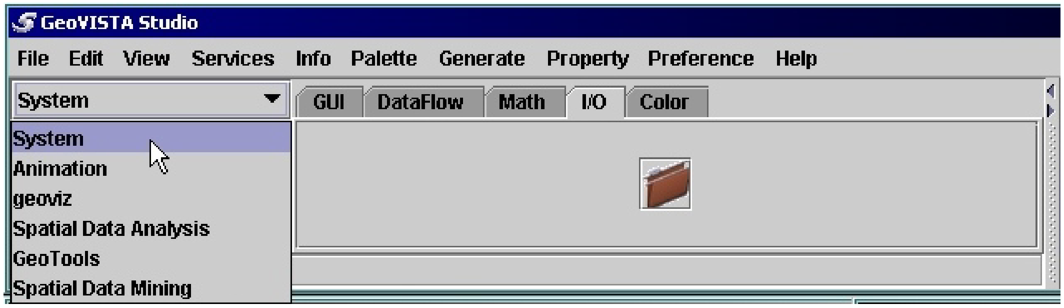
If you wish, you may follow this brief set of steps to create a simple design. This design will create a simple fixed-row matrix of maps and scatterplots.

You can turn off the gridlines in the DesignBox to make the design environment a little less confusing. To do this, right click on a blank space in the DesignBox and click "View ->" "Hide Grid."

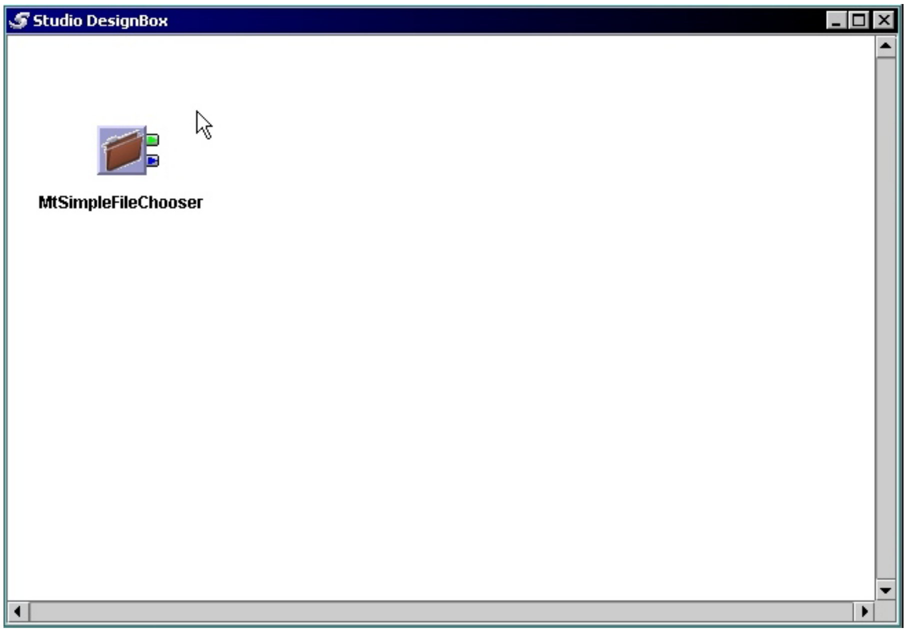
The **finished** design will look like this:



To begin, add the FileChooser bean by selecting the "System" palette and clicking on the "I/O" tab.


GeoVISTA Studio
File Edit View Services Info Palette Generate Property Preference Help
System
GUI DataFlow Math I/O Color
System
Animation
geoviz
Spatial Data Analysis
GeoTools
Spatial Data Mining

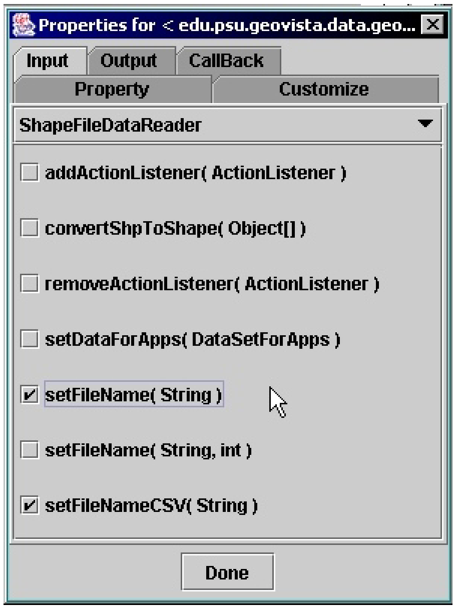
Click the icon for the FileChooser bean and place an instance of it in your DesignBox by clicking anywhere inside the DesignBox window.



Now that you've done this, you need to add the rest of the beans required for this design. From the "geoviz" palette, select and add **ShapeFileDataReader** , **ShapeFileProjection** , **ShapeFileToShape** , **CoordinationManagerGUI** , **MapMatrixElement** , **ScatterPlot** , and **FixedRowMatrix** beans to your DesignBox window.

To make your first connection, you need to enable the proper input ports on the ShapeFileDataReader. Each port represents a public "method," or action the bean can take. Right click on the ShapeFileDataReader bean and select the "Property" menu item. A dialog box will appear.

Make sure the selections shown are selected in your property box, and deselect any others that might already exist. The checked boxes represent actions the bean may be required to take in the current application. The unchecked boxes represent actions that would be used in other contexts.



ShapeFileDataReader

□ addActionListener(ActionListener)

□ convertShpToShape(Object[])

□ removeActionListener(ActionListener)

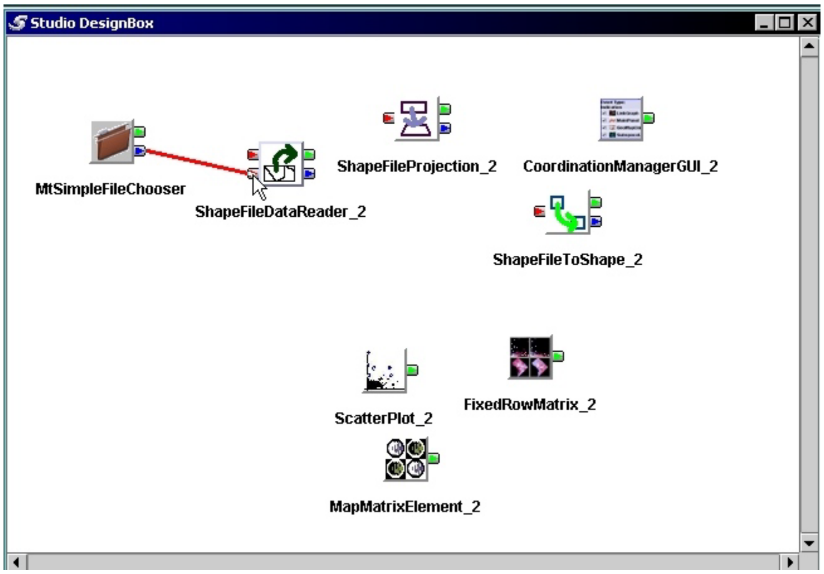
□ setDataForApps(DataSetForApps)

☑ setFileName(String)

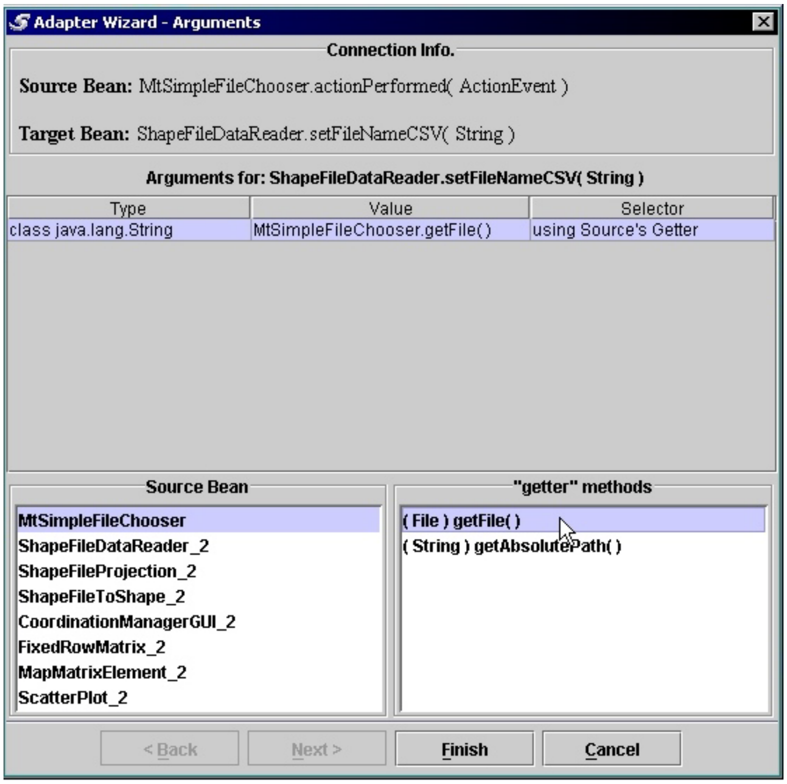
□ setFileName(String, int)

☑ setFileNameCSV(String)

Now you can make your first connection! Click and drag your mouse from the blue output of the FileChooser to either of the red inputs of the ShapeFileDataReader.



When you let go, a dialog will appear to ask you how you'd like the two beans to connect. Select the "getter" method that says (File)getFile().This means that the file chooser will pass a "File" object to the ShapeFileDataReader when asked. Then click the "Finish" button to complete your connection. Do the exact same thing again from the output of the FileChooser to the other input of the ShapeFileDataReader. By accomplishing this, you have now enabled your design to accept shapefiles with either .dbf or .csv attribute tables. The ShapeFileReader currently supports two input formats and now has a port for each.



Connection Info.

Source Bean: MtSimpleFileChooser.actionPerformed(ActionEvent)

Target Bean: ShapeFileDataReader.setFileNameCSV(String)

Arguments for: ShapeFileDataReader.setFileNameCSV(String)

Type	Value	Selector
class java.lang.String	MtSimpleFileChooser.getFile()	using Source's Getter

Source Bean

- MtSimpleFileChooser
- ShapeFileDataReader_2
- ShapeFileProjection_2
- ShapeFileToShape_2
- CoordinationManagerGUI_2
- FixedRowMatrix_2
- MapMatrixElement_2
- ScatterPlot_2

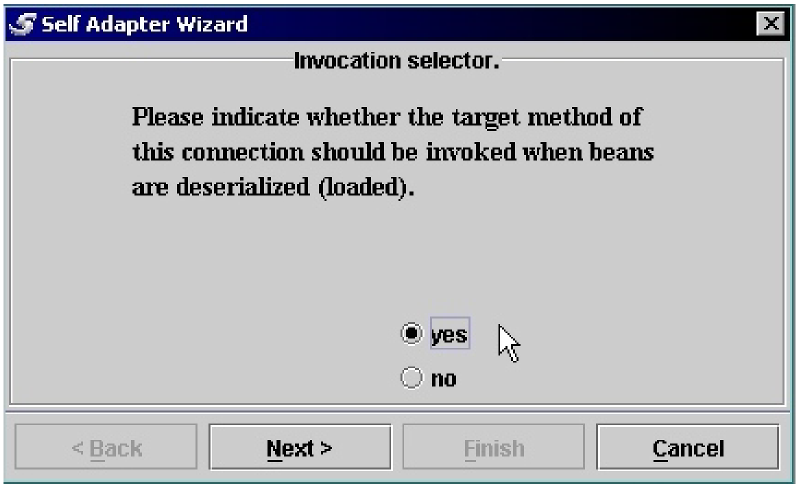
"getter" methods

- (File) getFile( )
- (String) getAbsolutePath( )

Next, connect the blue output of the ShapeFileDataReader bean to the red input of the ShapeFileProjection bean. Use the (Object[]) getDataSet() “getter” for this connection. The projection bean creates an Equidistant Conic projection based on the bounding box of the shapefile it’s given. This projection works well for areas at least several hundred miles wide, less so for smaller areas.

Similarly, connect the output of the ShapeFileProjection bean to the input of the ShapeFileToShape bean. Use the (Object[]) getOutputDataSet() “getter” for this connection.

Now you need to enable an input for the CoordinationManagerGUI bean. This bean coordinates the behavior of other beans. Right click on the bean icon, select the “Property” menu item, and check the box under the Input tab that says addBean(Object). This will make a red input appear. Connect the green “this” communicator from the ShapteFileToShape bean to this input. This will cause a somewhat confusing dialog to appear.

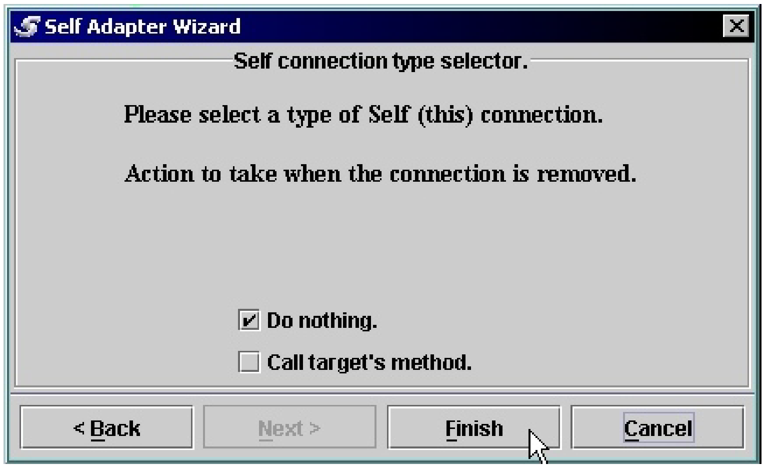


Invocation selector.

Please indicate whether the target method of this connection should be invoked when beans are deserialized (loaded).

● yes
○ no

Select “yes” and click on “Next >.” This will bring up another dialog:



Self connection type selector.

Please select a type of Self (this) connection.

Action to take when the connection is removed.

☑ Do nothing.
☐ Call target's method.

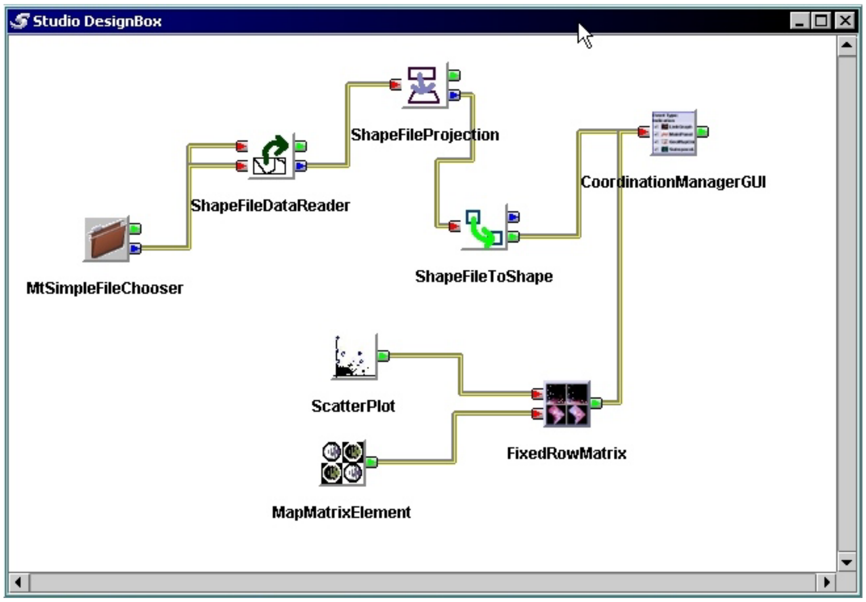
Select “Do nothing” and click “Finish.” These options are designed to control how beans are loaded and what should happen if they are disconnected.

Next you need to add two inputs to the FixedRowMatrix bean. Right click on the bean icon, select the “Property” menu item and enable SetElementClass1(Object) and SetElementClass2(Object). This exposes the methods that set the types of graphs inserted in the matrix. It also tells the matrix how many input elements to expect, and since you are creating a design with a map element and a scatterplot element, you need two.

Now, connect the “this” green connector from the ScatterPlot bean to the SetElementClass1(Object) input on the FixedRowMatrix bean. This inserts scatterplots into the first row of graphs in the FixedRowMatrix bean. To see which input is which, roll over the input with your mouse and wait for the tooltip to appear. When you connect the ScatterPlot to the FixedRowMatrix it will show you the same “Self Adapter Wizard” that you just saw above. Follow the same steps and complete your connection.

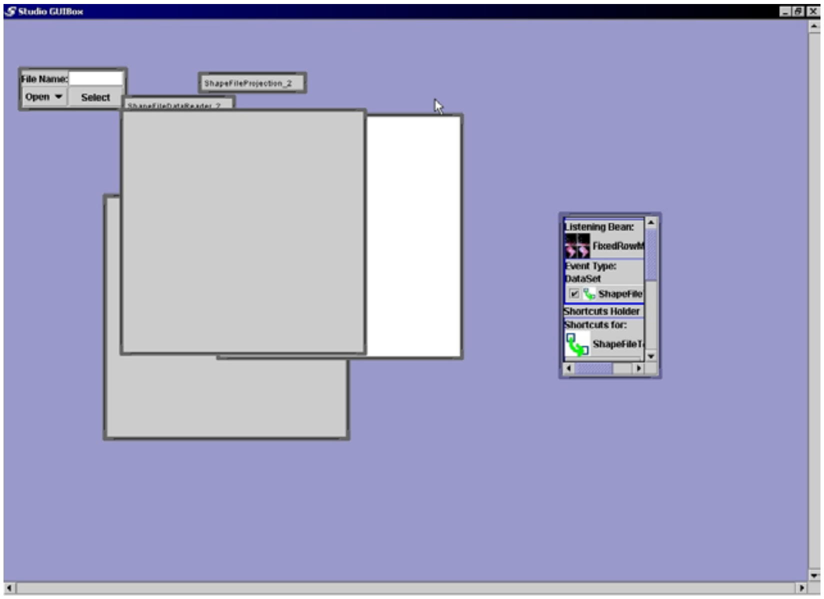
Repeat the process with the MapMatrixElement bean, connecting it to SetElementClass2(Object) on the FixedRowMatrix bean.

Congratulations - You've made your first design! It should look like this:



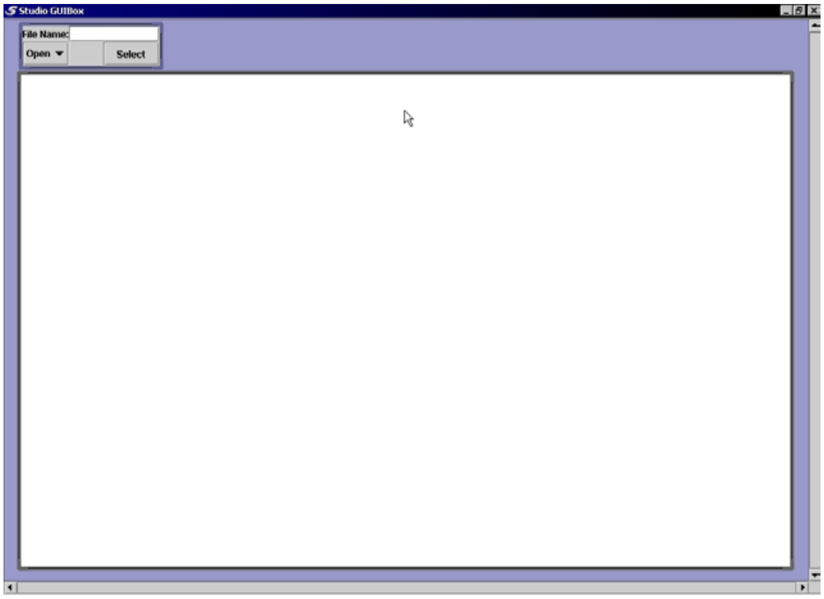
But you're not quite finished…

So far the *Studio* GUIBox has been neglected. Now is the time when you should maximize the GUIBox window and begin arranging your application components. Your initial GUI will look something like this:



In this design, the only boxes you really need to see are the FixedRowMatrix and the FileChooser. The other elements can be hidden or tucked away into the corner. The large white box is the FixedRowMatrix. If you are unsure what bean is represented by what box, go back to the DesignBox window and click on the bean you're interested in. It will highlight in the GUIBox.

Here's what you could shoot for (but feel free to be creative, of course!):



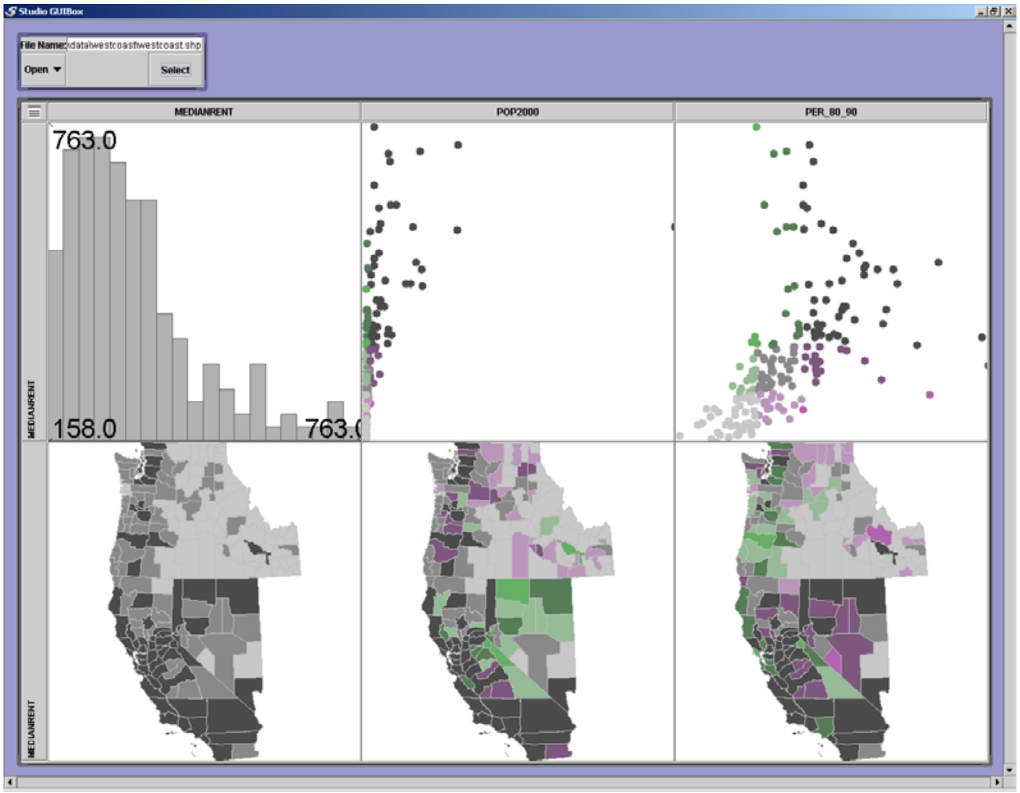
In this rather simple design, you only have two elements that need space in the GUI. In more complicated designs, the task of designing a GUI can be much more involved.

Now it's time to save your work. Click "File" and "Save Design" in the main application window.

- *An important caveat to the current* <u>*Studio*</u> *build is that designs are not portable to other computers if you save them while data is loaded in the GUI. So, if you wish to build your own design, you should save iteratively and test after saving.* We're working on rectifying this problem!

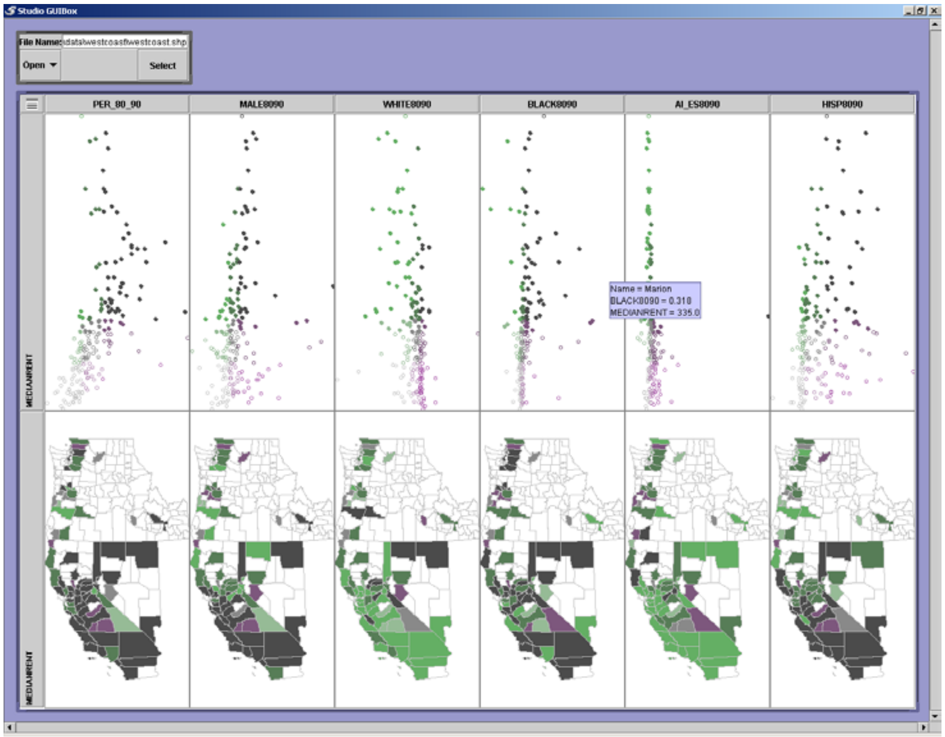
Now you should try your design by loading some data.

Here is the sort of visualization you should see if everything is working:



File Name: \data\westcoast\westcoast.shp

Studio is designed for you to explore data, so go ahead and try looking at different attributes, select some areas on the scatterplots and/or maps, and see what you can find.



File Name: \data\westcoast\westcoast.shp

Name = Marion
BLACK8090 = 0.310
MEDIANRENT = 335.0

Tell Us What You Think!

We'd love to hear from you about your experience using GeoVISTA *Studio*. Have a problem, comments, or questions? Email us at acr181@psu.edu .