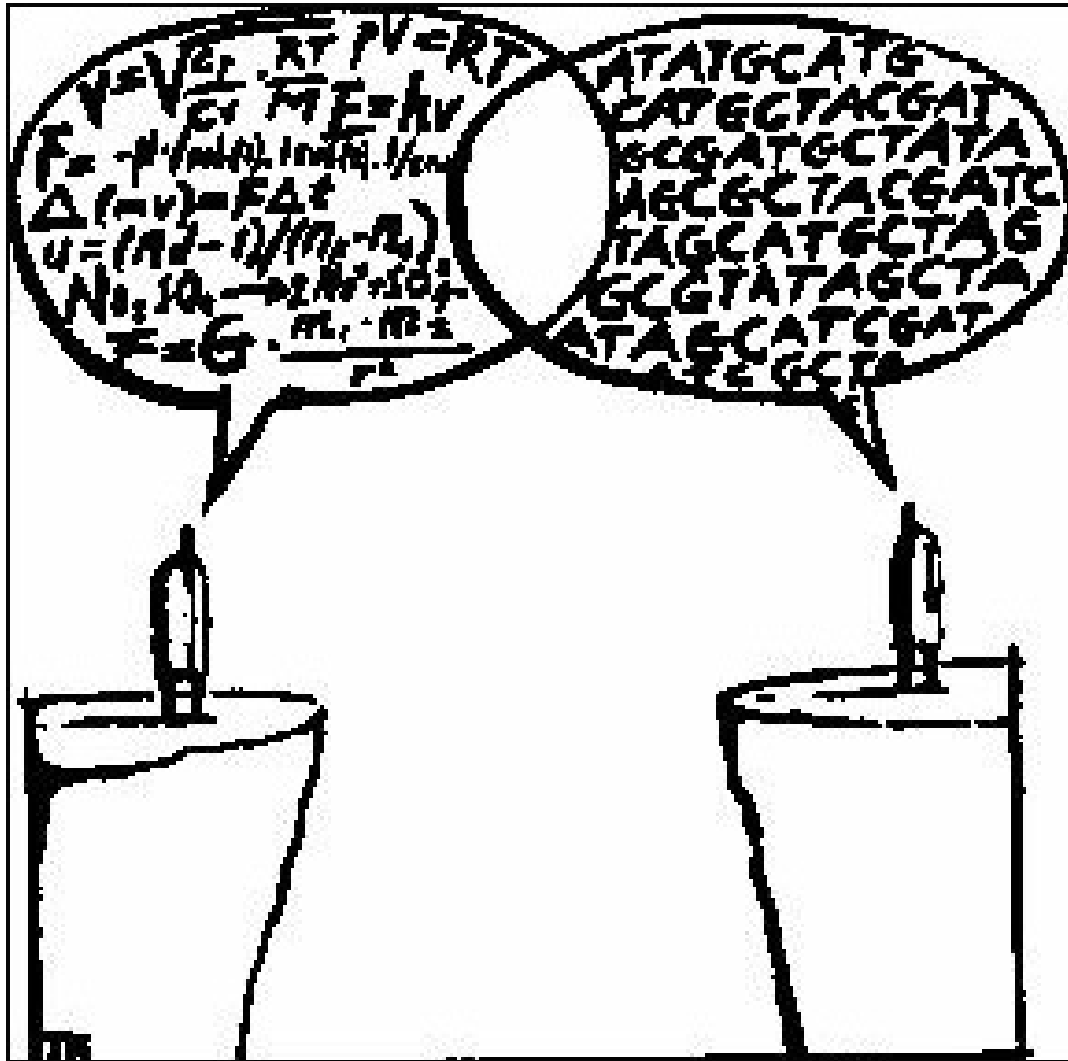




Visual programming for R

Anup Parikh (anup@red-r.org)

Kyle Covington (kyle@red-r.org)



Red-R Motivation

- Hide the code complexity and improve readability
- Create a more interactive platform for data exploration
- Improve data and analysis sharing between users
- Provide a community repository of analysis pipelines

Architecture

Red-R

- Visual Programming
- Interactive Graphics
- Analysis interpretability and sharing

Qt UI Framework

Python

RPy

R

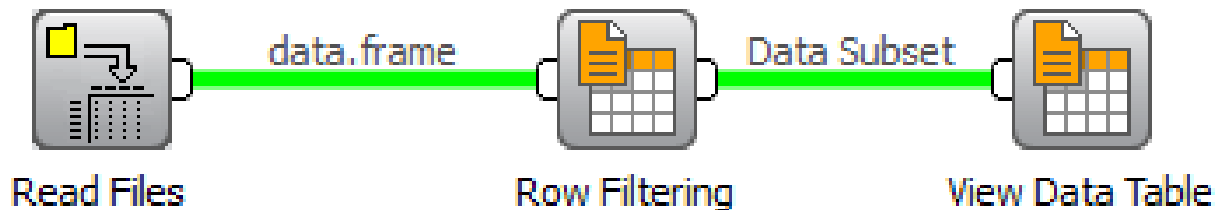
R Graphics
And
User interfaces

Red-R Motivation

- Hide the code complexity and improve readability
- Create a more interactive platform for data exploration
- Improve data and analysis sharing between users
- Provide a community repository of analysis pipelines

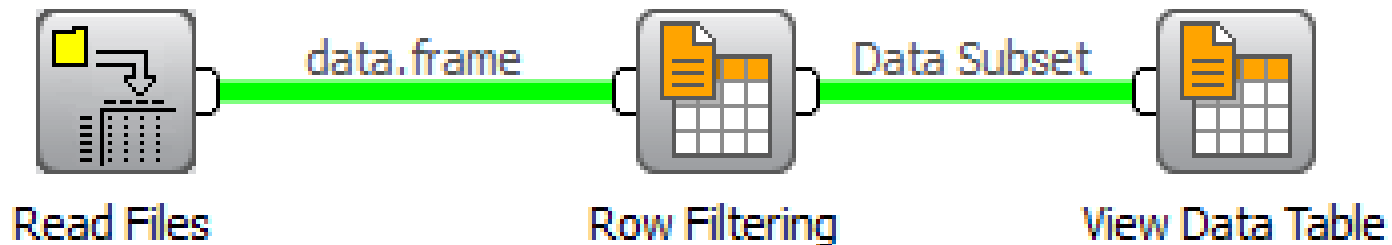
Visual Programming

- Visual programming interface
 - Analysis is performed by linking a series of widgets together
- Widgets correspond to R function
 - Read, manipulate or visualize data

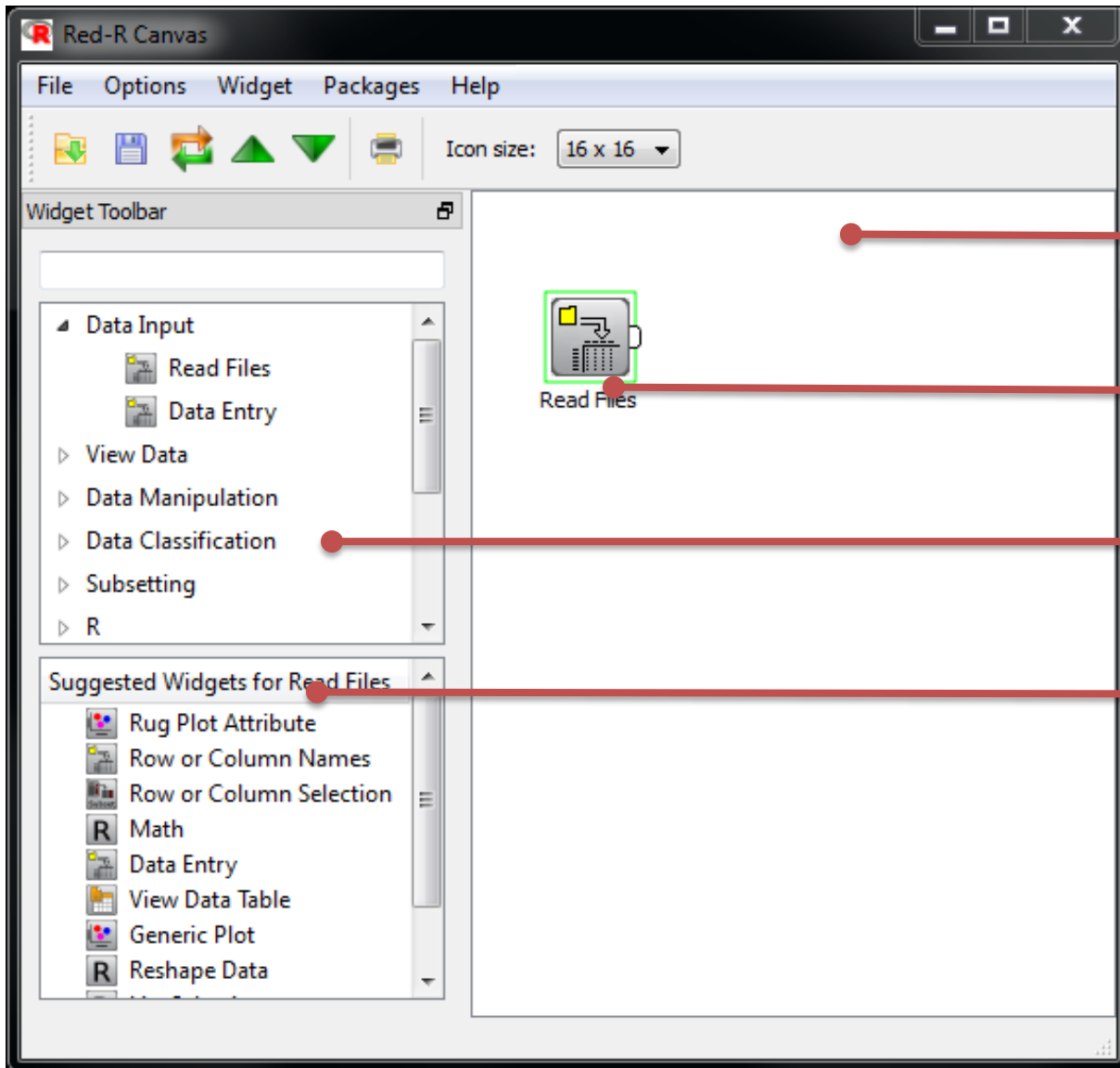


R vs. Red-R

```
1 diff_data = read.csv('diff_data.csv')
2 filtered = diff_data[diff_data$limma.pval.KAWvsBSW < .05
3 & abs(diff_data$FC.KAWvsBSW) >= 2 & diff_data$samplesWithExpression > 0]
4 write.csv(filtered, file='filtered.csv')
5
```



Red-R Overview



Canvas

Widget

All Widget

Widget
Suggestions

Widget

The screenshot displays the Red-R Canvas interface with the Read Files widget selected. The widget's configuration panel is open, showing the following settings:

- Load File:** iris.txt
- File Type:** Text
- Column Separator:** Tab
- Row and Column Names:** Column Headers checked, Select Row Names: NULL
- Other Options:** fill, strip.white, blank.lines.skip (checked), allowEscapes, stringsAsFactors

The widget's output is displayed in a table with the following data:

Rownames	row.names	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	1	5.1	3.5	1.4	0.2
2	2	4.9	3.0	1.4	0.2
3	3	4.7	3.2	1.3	0.2
4	4	4.6	3.1	1.5	0.2
5	5	5.0	3.6	1.4	0.2
6	6	5.4	3.9	1.7	0.4
7	7	4.6	3.4	1.4	0.3

The widget's documentation panel provides the following information:

Documentation: Reads files from a text file and brings them into RedR. These files should be like a table and should have values that are separated either by a tab, space, or comma. You can use the scan feature to scan a small part of your data and make sure that it is in the correct format. You can also set a column to represent the

Notes: Add your own note here

R code executed in this widget:

```
> filename_1_1278709200.56 =  
"C:/Users/anup/Documents/red/develop/red-trunk/iris.txt"  
> dataframe_org_1_1278709200.56 <-  
read.table(filename_1_1278709200.56, header =  
TRUE, sep = "\t", quote = "\"", colClasses = NA,  
row.names = NULL, skip = 0, nrows =  
10, blank.lines.skip = TRUE, dec = '.')  
> colnames(dataframe_org_1_1278709200.56)
```

Processing complete.

Widget

The screenshot displays the Red-R Canvas interface. The 'Read Files' widget is highlighted with a red border. The widget's configuration panel on the left includes the following settings:

- Load File: iris.txt
- File Type: Text
- Column Separator: Tab
- Row and Column Names: Column Headers checked, Select Row Names set to NULL
- Other Options: blank.lines.skip checked, # Lines to Scan: 10, # Lines to Skip: 0

The central data table shows the following content:

Rownames	row.names	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	1	5.1	3.5	1.4	0.2
2	2	4.9	3.0	1.4	0.2
3	3	4.7	3.2	1.3	0.2
4	4	4.6	3.1	1.5	0.2
5	5	5.0	3.6	1.4	0.2
6	6	5.4	3.9	1.7	0.4
7	7	4.6	3.4	1.4	0.3

The 'Other Options' section at the bottom of the widget configuration includes a list of data types for each column:

- row.names: factor
- Sepal.Length: numeric
- Sepal.Width: numeric
- Petal.Length: numeric
- Petal.Width: numeric
- Species: factor

The 'Documentation' panel on the right provides a description of the widget's function and the R code executed in this widget:

```
> filename = "C:/Users/ananup/Documents/red/develop/red-trunk/iris.txt"
> dataframe_1_1278709200.56 <- read.table(filename_1_1278709200.56, header = TRUE, sep = "\t", quote = "\"", colClasses = NA, row.names = NULL, skip = 0, nrows = 10, blank.lines.skip = TRUE, dec = '.')
> colnames(dataframe_1_1278709200.56)
```

At the bottom of the window, the status bar indicates 'Processing complete.' and buttons for 'Help', 'Notes', 'R Output', and 'Print' are visible.

Widget

The screenshot displays the Red-R Canvas interface with the Read Files widget selected. The widget's configuration panel is open, showing the following settings:

- Load File: iris.txt
- File Type: Text
- Column Separator: Tab
- Row and Column Names: Column Headers checked, Row Names set to NULL
- Other Options: blank.lines.skip checked, # Lines to Scan: 10

The widget's output is displayed in a table with the following data:

Rownames	row.names	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	1	5.1	3.5	1.4	0.2
2	2	4.9	3.0	1.4	0.2
3	3	4.7	3.2	1.3	0.2
4	4	4.6	3.1	1.5	0.2
5	5	5.0	3.6	1.4	0.2
6	6	5.4	3.9	1.7	0.4
7	7	4.6	3.4	1.4	0.3

The widget's documentation is visible on the right, and the R code executed in this widget is shown below:

```
> filename_1_1278709200.56 =  
"C:/Users/anup/Documents/red/develop/red-trunk/iris.txt"  
> dataframe_org_1_1278709200.56 <-  
read.table(filename_1_1278709200.56,  
TRUE, sep = "\t", quote = "\"", colClasses  
row.names = NULL, skip=0, nrows =  
10, blank.lines.skip=TRUE, dec = '.')  
> colnames(dataframe_org_1_1278709200.56)
```

Red callouts are present over the interface:

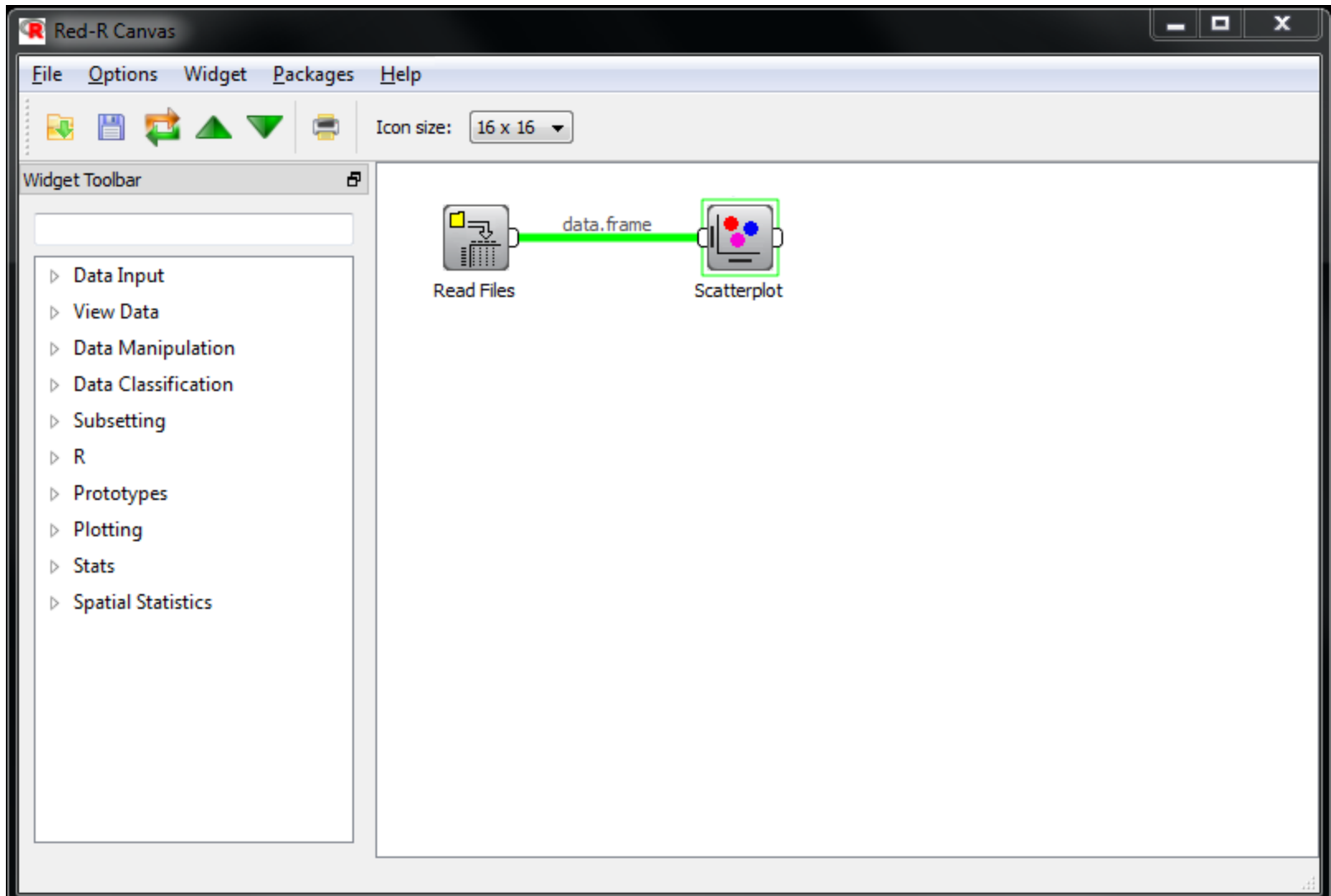
- Help**: A red callout box is positioned over the documentation text.
- Notes**: A red callout box is positioned over the Notes section.
- R code**: A red callout box is positioned over the R code output.

Buttons for Help, Notes, R Output, and Print are located at the bottom right of the widget window.

Red-R Motivation

- Hide the code complexity and improve readability
- Create a more interactive platform for data exploration
- Improve data and analysis sharing between users
- Provide a community repository of analysis pipelines

Creating a Workflow



Interactive Widgets

The screenshot displays the Red-R Canvas interface. At the top, the menu bar includes File, Options, Widget, Packages, and Help. Below the menu is a toolbar with icons for file operations and a dropdown for icon size (16 x 16). A 'Widget Toolbar' on the left lists various widget categories: Data Input, View Data, Data Manipulation, Data Classification, Subsetting, R, Prototypes, Plotting, Stats, and Spatial Statistics.

The main workspace shows a workflow: 'Read Files' widget connected to a 'data.frame' data object, which is then connected to a 'Scatterplot' widget. A 'Scatterplot' dialog box is open, showing the following settings:

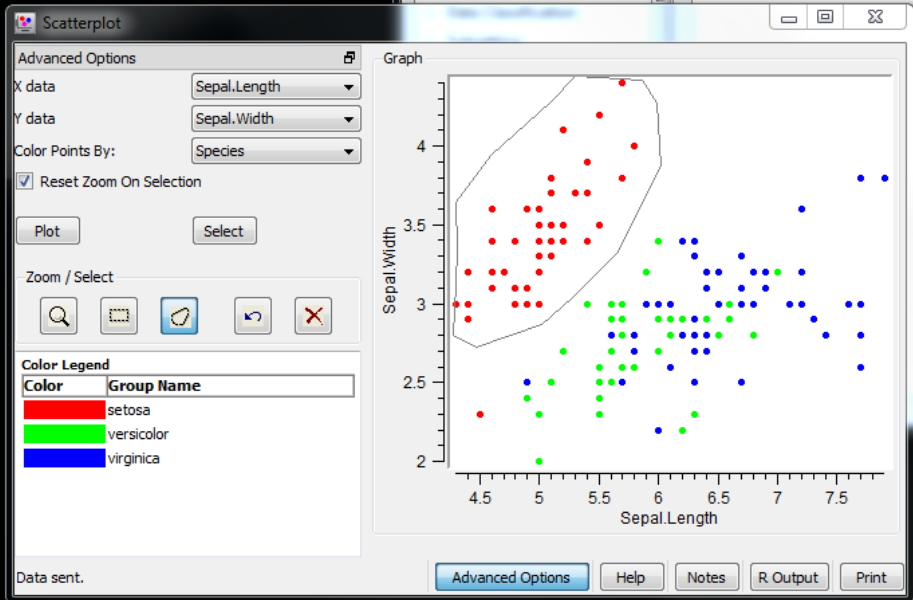
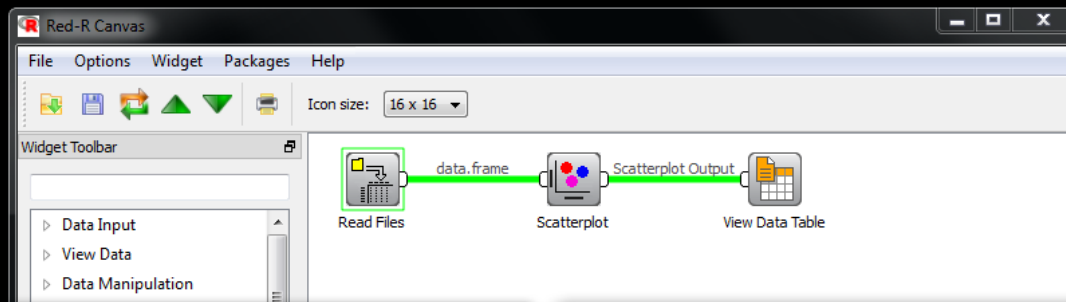
- Advanced Options:** X data: Sepal.Length, Y data: Sepal.Width, Color Points By: Species, Reset Zoom On Selection.
- Buttons:** Plot, Select.
- Zoom / Select:** Includes icons for zoom in, zoom out, pan, and reset.
- Color Legend:**

Color	Group Name
Red	setosa
Green	versicolor
Blue	virginica

The scatterplot graph shows Sepal.Width on the y-axis (ranging from 2 to 4) and Sepal.Length on the x-axis (ranging from 4.5 to 7.5). The data points are colored according to the legend, showing three distinct clusters: red (setosa), green (versicolor), and blue (virginica).

At the bottom of the dialog, there are buttons for 'Advanced Options', 'Help', 'Notes', 'R Output', and 'Print'. The status bar at the very bottom indicates 'Data sent.'

Interactive Workflows



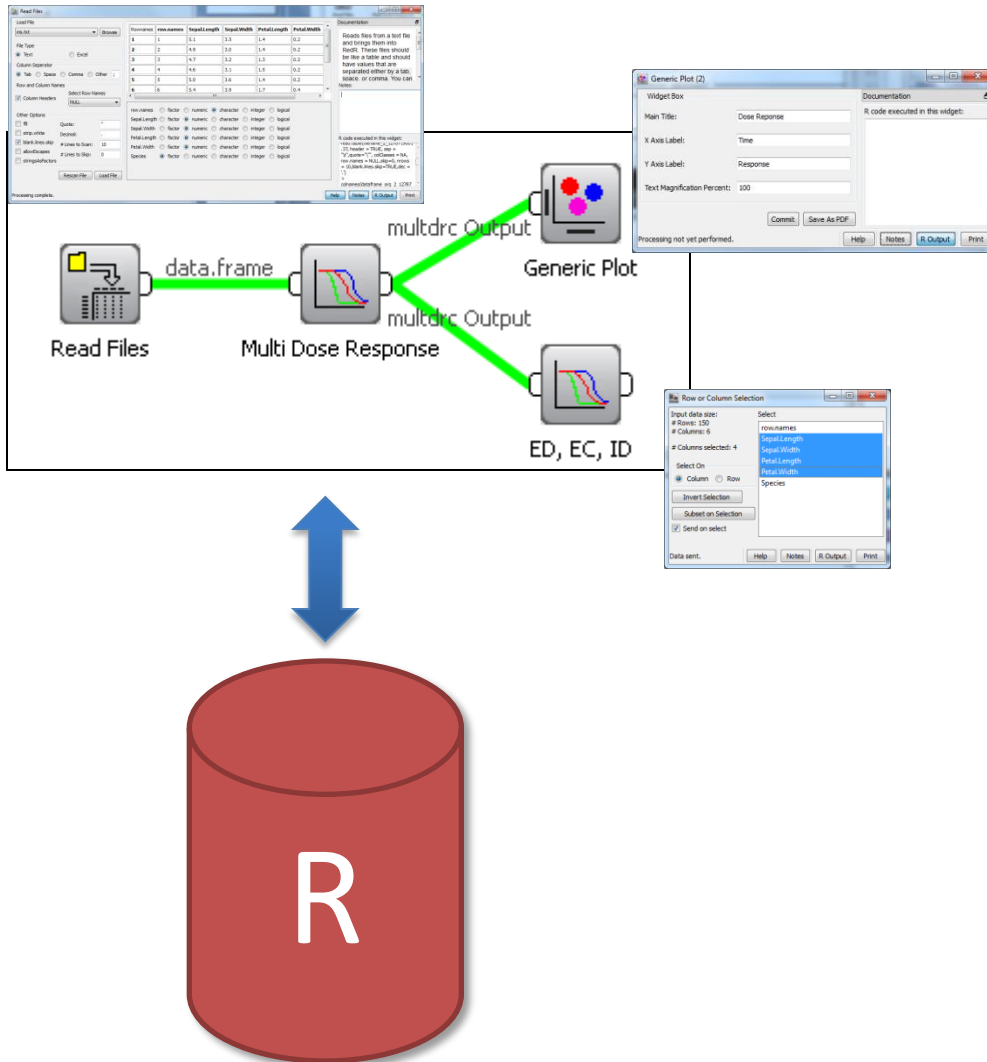
The View Data Table window displays a table of data with the following columns: row.names, Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species. The table contains 10 rows of data, with the first row highlighted.

row.names	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

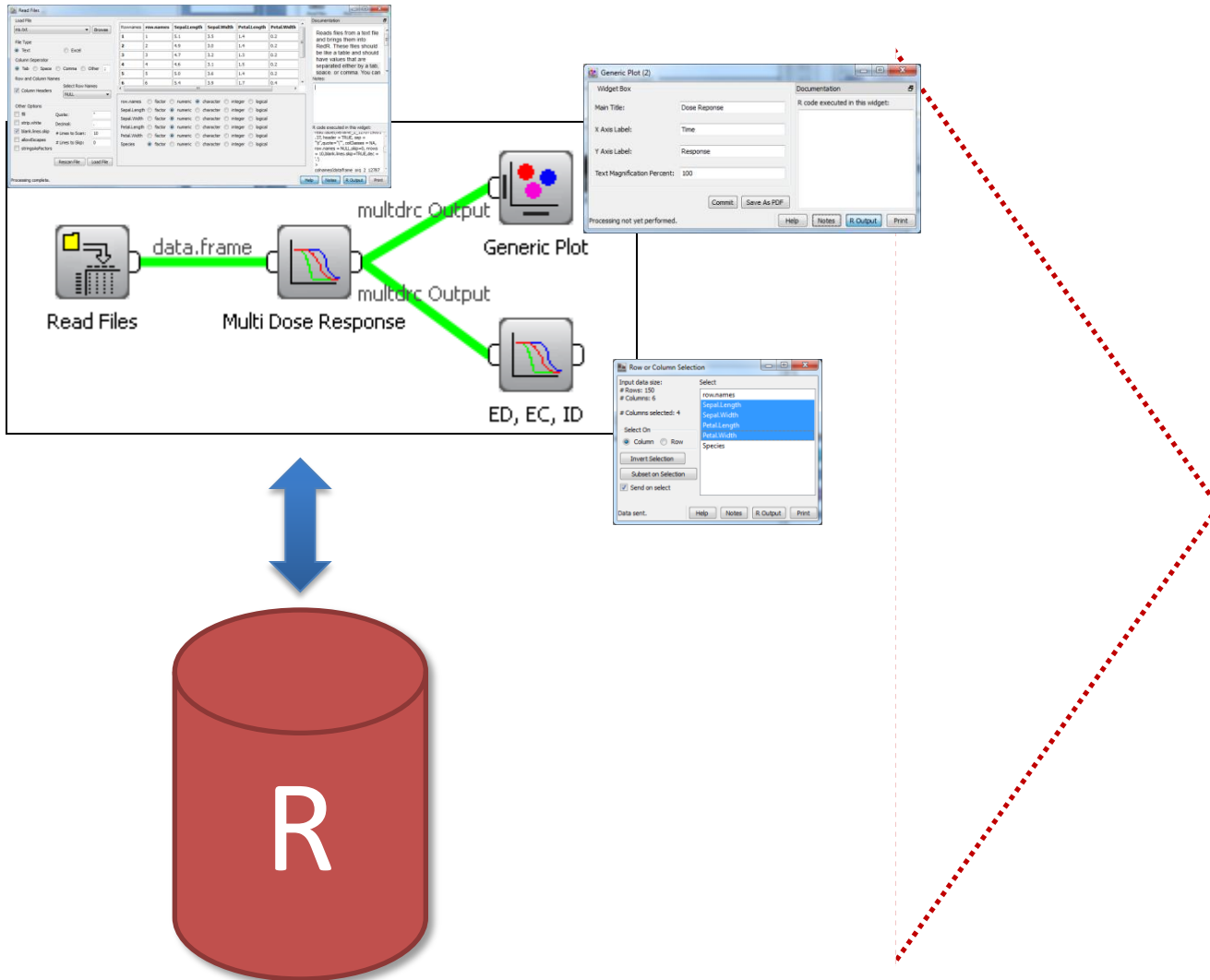
Red-R Motivation

- Hide the code complexity and improve readability
- Create a more interactive platform for data exploration
- **Improve data and analysis sharing between users**
- Provide a community repository of analysis pipelines

Data Sharing



Data Sharing



One Shareable File
Workflow Parameters
Outputs
Notes

Import Existing R Sessions

The screenshot displays the Red Canvas interface with a workflow diagram. The workflow starts with a 'Load R Session' widget, which branches into three parallel paths:

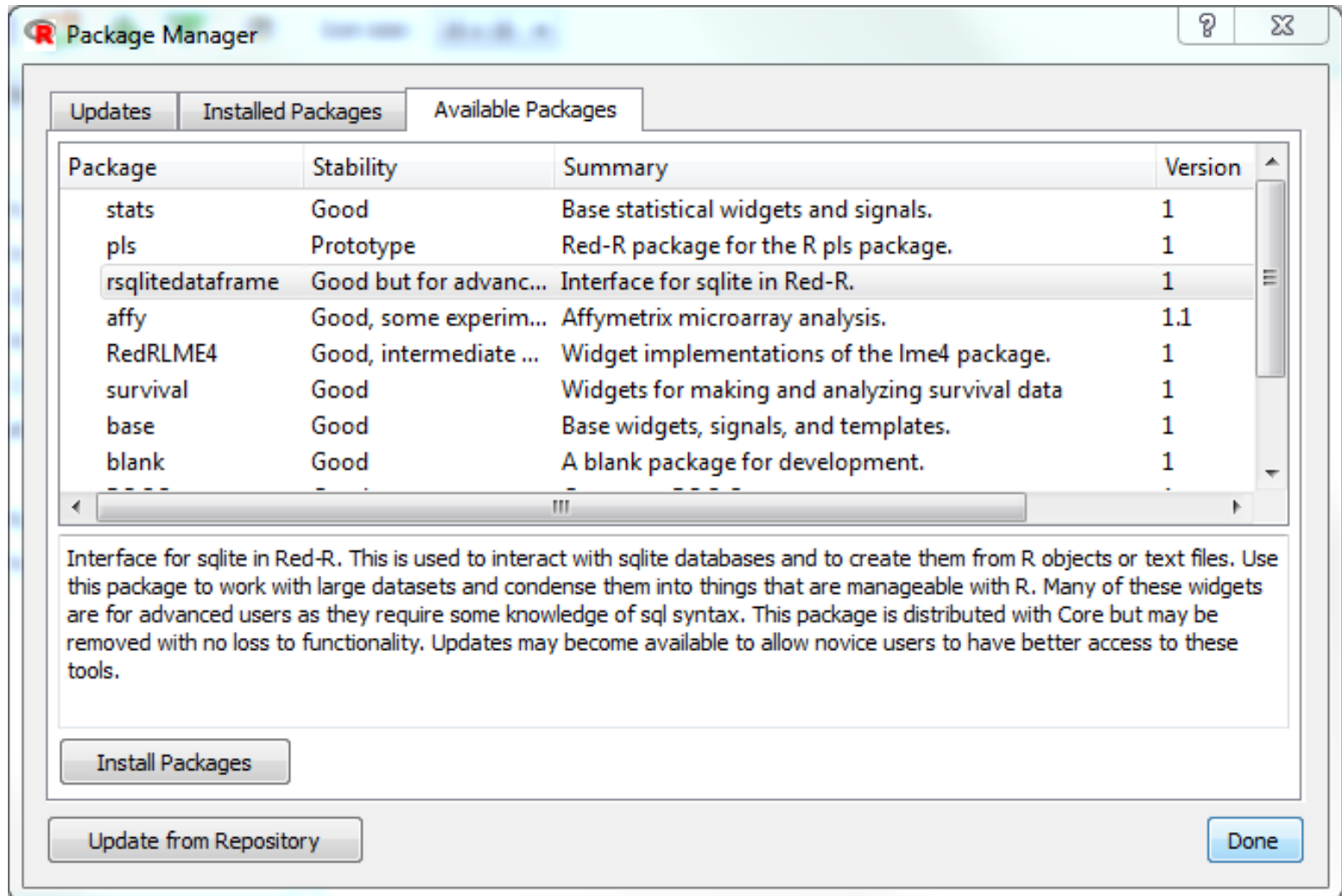
- Top Path:** 'Load R Session' (R) → 'R Variable Selection' (R) → 'R Data Frame' → 'Row Filtering' → 'Data Subset' → 'View Data Table'.
- Middle Path:** 'Load R Session' (R) → 'R Variable Selection (2)' (R) → 'R Data Frame' → 'Scatterplot' → 'Scatterplot Output' → 'View Data Table (2)'.
- Bottom Path:** 'Load R Session' (R) → 'R Variable Selection (3)' (R) → 'R Data Frame' → 'Row or Column Selection' → 'Data Table' → 'Correlation/Variance'.

The 'Widget Toolbar' on the left lists various R-related widgets, including 'View R Output', 'Widget Maker', 'Remove NA', 'Summary', 'Load R Session', 'R Variable Selection', 'Rename Rows or Columns', 'Dummy', 'R Executor', and 'Sort'. The 'Load R Session' widget is highlighted in the toolbar.

Red-R Motivation

- Hide the code complexity and improve readability
- Create a more interactive platform for data exploration
- Improve data and analysis sharing between users
- Provide a community repository of analysis pipelines

Community Repository: Packages



The screenshot shows the R Package Manager window with the 'Available Packages' tab selected. The window title is 'Package Manager'. The interface includes a table of available packages, a scrollable area for the selected package's details, and buttons for 'Install Packages', 'Update from Repository', and 'Done'.

Package	Stability	Summary	Version
stats	Good	Base statistical widgets and signals.	1
pls	Prototype	Red-R package for the R pls package.	1
rsqldataframe	Good but for advanc...	Interface for sqlite in Red-R.	1
affy	Good, some experim...	Affymetrix microarray analysis.	1.1
RedRLME4	Good, intermediate ...	Widget implementations of the lme4 package.	1
survival	Good	Widgets for making and analyzing survival data	1
base	Good	Base widgets, signals, and templates.	1
blank	Good	A blank package for development.	1

Interface for sqlite in Red-R. This is used to interact with sqlite databases and to create them from R objects or text files. Use this package to work with large datasets and condense them into things that are manageable with R. Many of these widgets are for advanced users as they require some knowledge of sql syntax. This package is distributed with Core but may be removed with no loss to functionality. Updates may become available to allow novice users to have better access to these tools.

Install Packages

Update from Repository

Done

Community Repository: Templates

The screenshot shows a web browser window displaying the 'Templates' page on the Red-R website. The browser's address bar shows the URL 'www.red-r.org/documentation/templates'. The website's header includes the Red-R logo and the tagline 'visual programming for R'. A navigation menu is visible with links for HOME, DOWNLOADS, DOCUMENTATION, DEVELOPMENT, CONTACT, and FORUMS. Below the navigation, there are sub-links for Packages and Templates. The main content area is titled 'TEMPLATES' and contains a brief introduction: 'Templates hold a set of widgets that work together to perform a specific task.' followed by a link to 'Edit this entry.' The page features a grid of template cards, each with a title and a short description. On the right side, there are two sidebars: 'RECENT TEMPLATES' and 'TAG CLOUD'. The 'RECENT TEMPLATES' sidebar lists several templates, including 'Neural Net Plot and Summary', 'Boxplot', 'RDataFrame to Database', 'Connect and Query', and 'Principal Components Plotting'. The 'TAG CLOUD' sidebar displays a collection of tags, with 'Stats' being the most prominent.

RED R visual programming for R

Search: type and hit enter!

HOME DOWNLOADS DOCUMENTATION DEVELOPMENT CONTACT FORUMS

Packages Templates

TEMPLATES

Templates hold a set of widgets that work together to perform a specific task.

[Edit this entry.](#)

Neural Net Plot and Summary

This template makes a neural net from data read into the Schema using Read Files. The data is used to form a Neural Net using the Make Neural Net wi ...

Boxplot

This template reads in data using Read Files and plots the data using the RedR Matplotlib boxplot widget. Boxplots are printable and interactive. D ...

RDataFrame to Database

Reads in a file using Read Files and saves it to a database. This database will be in the Red-R session by default but can be saved to other locatio ...

Connect and Query

This template connects to an SQLite database and then allows querying of that database using the Run SQLite Query widget. The output can be used as ...

Principal Components Plotting

This template generates Principal Components fits to data. Data should be numeric (no text). If any

Spline Fit

This template generates a Spline fit to X and Y data. X and Y data are set using the List Selector

RECENT TEMPLATES

- [Neural Net Plot and Summary](#)
- [Boxplot](#)
- [RDataFrame to Database](#)
- [Connect and Query](#)
- [Principal Components Plotting](#)

TAG CLOUD

ANOVA Base **Bioinforma**
Correlation Database Data
Manipulation Dose Response Fil
Model loading Merge Microar
Non Parametric Parametric **PL**
Principal Components R ROC Sp
Stats Subset
View Data XYPlot

Community Repository: Templates

The screenshot shows two overlapping browser windows. The background window displays the 'Templates | Red-R' page with a navigation menu (HOME, DOWNLOADS, DOCUMENTATION) and a list of templates. The foreground window displays the 'Spline Fit | Red-R' page, which includes a search bar, a navigation menu (HOME, DOWNLOADS, DOCUMENTATION, DEVELOPMENT, CONTACT, FORUMS), and a detailed description of the Spline Fit template. The description states: 'This template generates a Spline fit to X and Y data. X and Y data are set using the List Selector widget from data read into the canvas using Read Files. The data is fit to a spline using the Spline Fit widget and plotted using the RedRplot widget. The spline fit is added to the plot after plotting. A summary of the spline fit is also given in the Summary widget.' Below the description, there is a 'Download Template' button and a diagram illustrating the data flow.

RELATED WIDGETS AND TEMPLATES

- [Spline Fit](#)
- [Principal Components Plotting](#)
- [Neural Net Plot and Summary](#)
- [Wilcoxon Test](#)
- [Survival Test](#)

Diagram illustrating the data flow for the Spline Fit template:

```
graph TD; ReadFiles[Read Files] -- data.frame --> LS1[List Selection]; ReadFiles -- data.frame --> LS2[\"List Selection (2)\"]; LS1 -- R Vector --> SF[\"Stats Spline Fit\"]; LS2 -- R Vector --> SF; LS2 -- R Vector --> RP[RedRplot]; SF -- spline Output --> Sum[Summary]; SF -- spline plot attribute --> RP;
```

Current Functionality

Base R functionality

- Read/View Data
- Subsetting
 - Merge/Intersect/Filter
- Manipulations
 - Math/Apply
- Plotting
 - Interactive Scatter Plot
 - Most R plots
- Stats
 - Parametric
 - Non-Parametric

Additional R packages

- Bioconductor microarray analysis
- Survival analysis
- Spatial Stats
- SQLite
- ROCR – ROC Curves
- Neural Nets
- LME4

Expanding Functionality

- How do you make it easier to transition from R to Red-R?

Expanding Functionality

- How do you make it easier to transition from R to Red-R?

The screenshot shows the R Widget Maker application window. It features a 'Function Info' tab and a 'Code' tab. The 'Function Info' section includes a 'Package:' field with a 'Load Package' button, a 'Function Name:' field with a 'Parse Function' button, and a 'GUI Args' field containing 'y x use method'. Below this is a table with columns: Name, Input Type, Required, Signal Class, Input class, Default, and Options. The table contains four rows of input parameters. At the bottom, there are checkboxes for 'Accept Inputs', 'Allow Output', and 'Show Output', an 'Output Class:' dropdown set to 'RDataFrame', and buttons for 'Generate Code' and 'Launch Widget'. A status bar at the bottom left says 'Processing not yet performed.' and the bottom right has buttons for 'Help', 'Notes', 'R Output', and 'Print'.

	Name	Input Type	Required	Signal Class	Input class	Default	Options
1	y	Connection Input	Optional	RDataFrame	lineEdit	NULL	
2	x	Connection Input	Optional	RDataFrame	lineEdit		
3	use	Widget Input	Optional	RDataFrame	radioBox	"everything"	ete.obs", "pairwise.complete.obs"]
4	method	Widget Input	Optional	RDataFrame	radioBox		[pearson', 'kendall', 'spearman']

Expanding Functionality

The screenshot shows the R Widget Maker interface. The top window, titled "Widget Maker", has two tabs: "Function Info" and "Code". The "Code" tab is active, displaying the following R code:

```
import libraries.base.signalClasses as signals

class RedRcor(OWRpy):
    settingsList = []
    def __init__(self, parent=None, signalManager=None):
        OWRpy.__init__(self, parent, signalManager, "cor", wantMainArea = 0, resizingEnabled = 1)
        self.RFunctionParam_y = ''
        self.RFunctionParam_x = ''
        self.inputs = [("y", signals.RDataFrame.RDataFrame, self.processy), ("x", signals.RDataFrame.RDataFrame, se

        self.RFunctionParamuse_radioBox = redRGUI.radioButton(self.controlArea, label = "use:", buttons = ["everything"],
        self.RFunctionParammethod_radioBox = redRGUI.radioButton(self.controlArea, label = "method:", buttons = [""],
        redRGUI.button(self.bottomAreaRight, "Commit", callback = self.commitFunction)

    def processy(self, data):
        if data:
            self.RFunctionParam_y=data.getData()
            #self.data = data
```

Below the code editor, the status bar indicates "Processing not yet performed." and contains buttons for "Help", "Notes", "R Output", and "Print".

The bottom window, also titled "Widget Maker", has a status bar with "Processing not yet performed." and buttons for "Help", "Notes", "R Output", and "Print". The main area of this window contains several controls:

- Buttons: "Accept Inputs", "Allow Output" (with a checked checkbox), "Show Output" (with an unchecked checkbox), "Generate Code", and "Launch Widget".
- Output Class: A dropdown menu currently set to "RDataFrame".

Highlights

- Reduced learning curve for access to R functionality
- Analysis methods easier to read and understand and share
 - Hopefully leads to analysis reproducibility
- Increase productivity with interactivity



<http://www.red-r.org>

