



# Using R in Kepler

Chad Berkley for Dan Higgins

National Center for Ecological Analysis and Synthesis

UC Santa Barbara

<http://www.kepler-project.org>



# What is 'R' ?

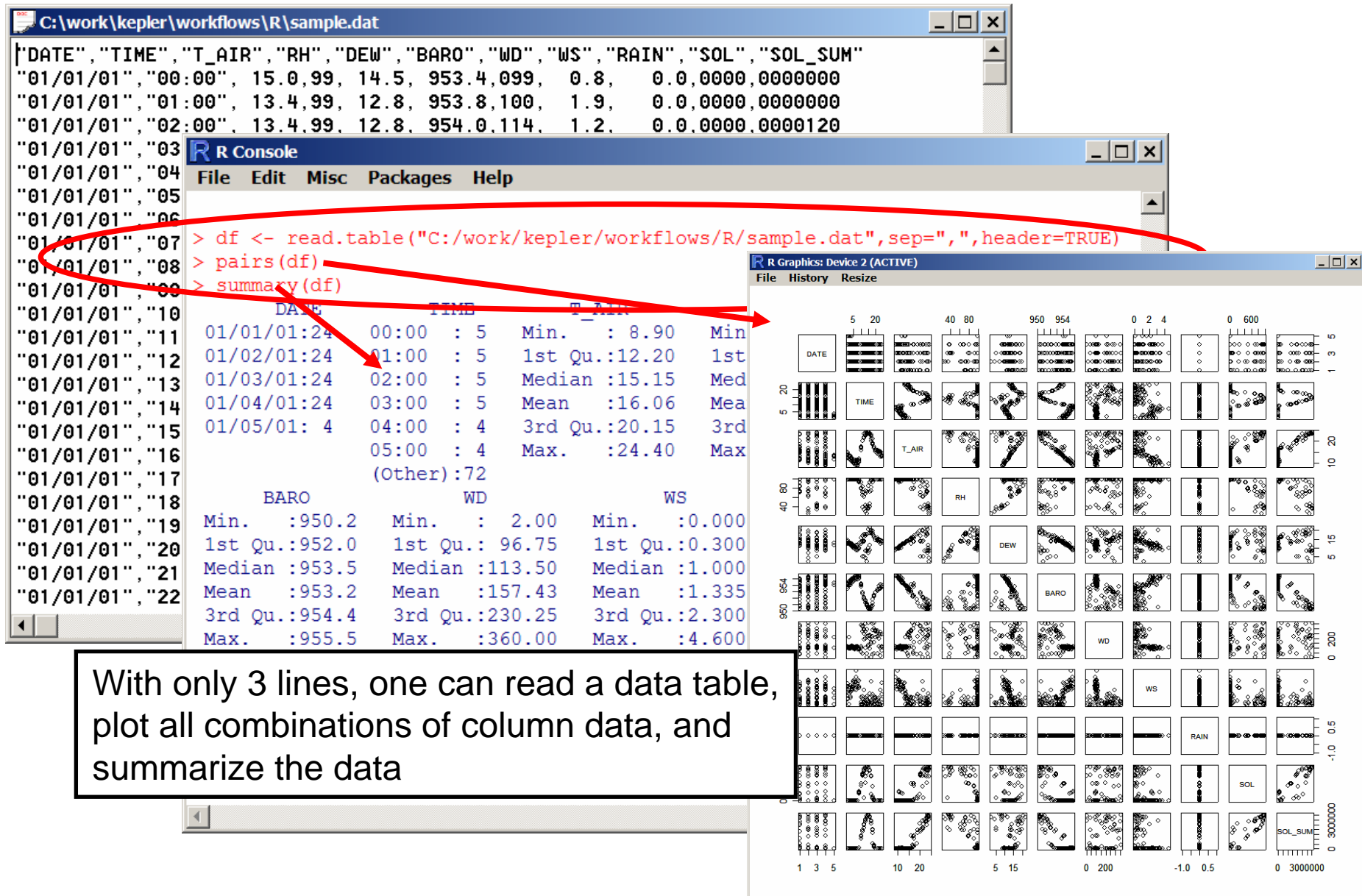
*"R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R. "*

*"R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity. "*

From the R Project Web page - <http://www.r-project.org/>



# R Example

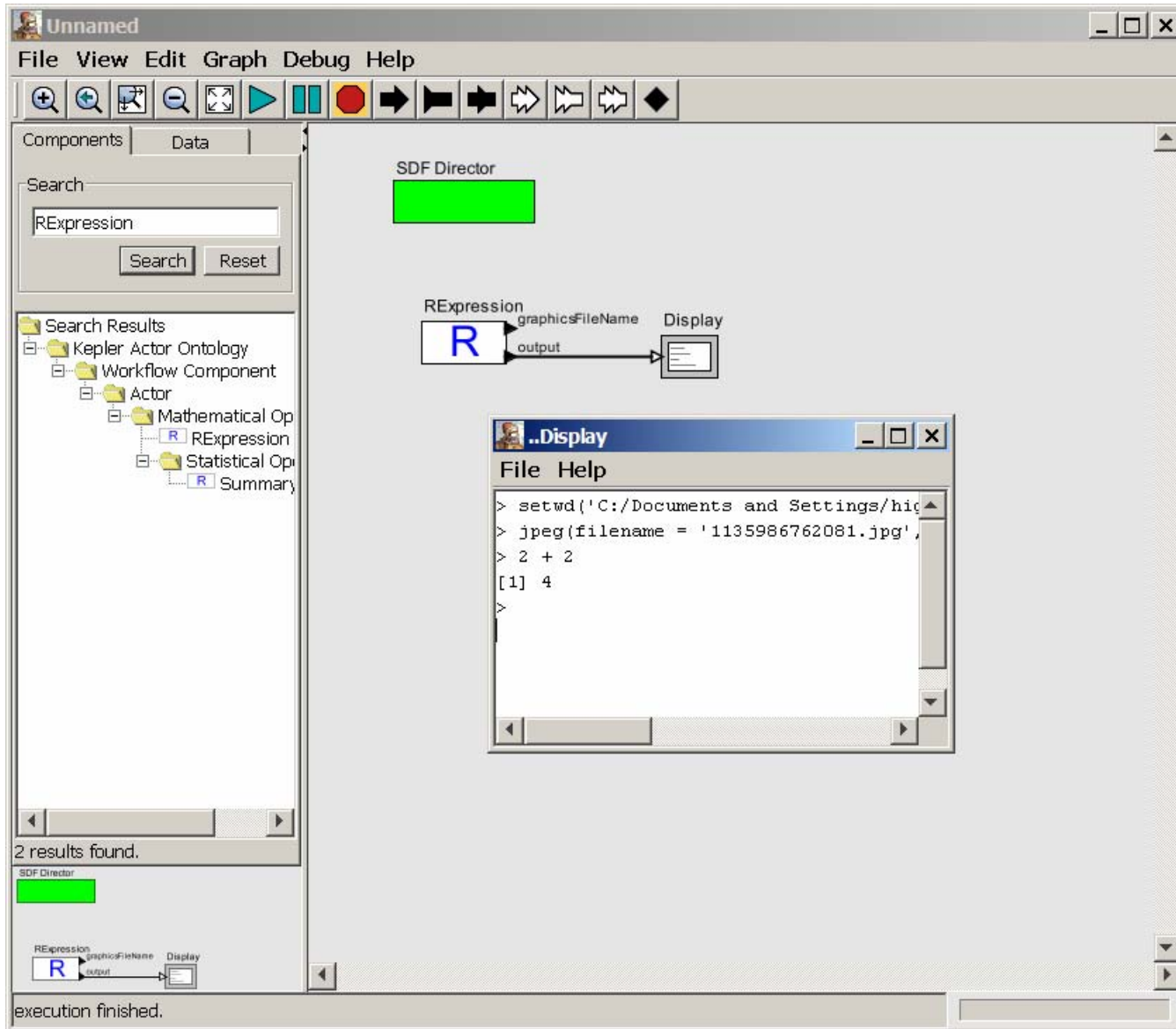




## Kepler and R

- R language has many similarities to the Kepler expression language
- R language emphasizes operations on vectors, matrices, and tables ('data frames') rather than scalars. (This eliminates many explicit looping statements)
- Many detailed statistical operations and data manipulation routines already exist in R
- R has ability to create sophisticated graphic displays
- Being able to call R routines from Kepler greatly simplifies many workflows

# Simple R Workflow



The screenshot shows the SEEK software interface. The main window is titled "Unnamed" and has a menu bar with "File", "View", "Edit", "Graph", "Debug", and "Help". Below the menu bar is a toolbar with various icons. On the left side, there is a "Components" panel with a "Search" box and a "Search Results" list. The "Search Results" list shows a tree structure with "Kepler Actor Ontology" as the root, followed by "Workflow Component", "Actor", "Mathematical Op", "RExpression", "Statistical Op", and "Summary". The "RExpression" component is selected. Below the "Search Results" list, it says "2 results found." and "SDF Director". In the main workspace, there is a workflow diagram. It starts with an "RExpression" actor (a box with a blue "R" logo) connected to a "Display" actor (a box with a document icon). The connection is labeled "graphicsFileName" and "output". Above the "RExpression" actor is a green rectangular box labeled "SDF Director". Below the workflow diagram, there is a terminal window titled "..Display" with a "File" and "Help" menu. The terminal shows the following R code and output:

```
> setwd('C:/Documents and Settings/hic')
> jpeg(filename = '1135986762081.jpg',
> 2 + 2
[1] 4
>
```

At the bottom of the main window, it says "execution finished."

Just drag an RExpression actor to the work area, add a director, and connect the output to a display

Result is the same as one sees running the R script from the command line

# RExpression Actor Parameters

**Edit parameters for RExpression** [X]

? R function or script:

2 + 2

R working directory: C:\Documents and Settings\higgins

save or not: --no-save

graphicsOutput: ☒

Number of X pixels in image: 480

Number of Y pixels in image: 480

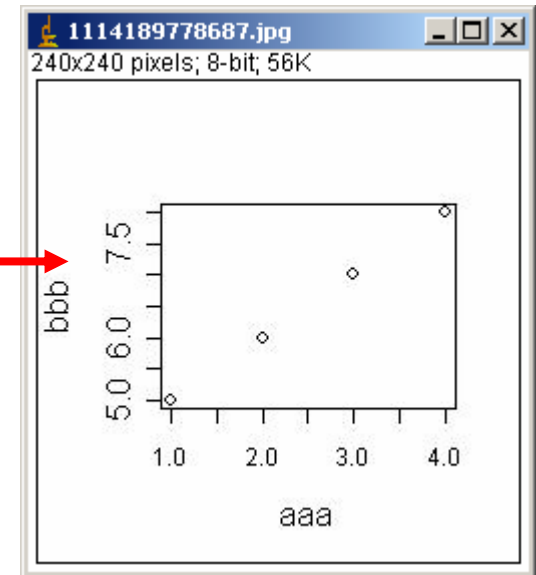
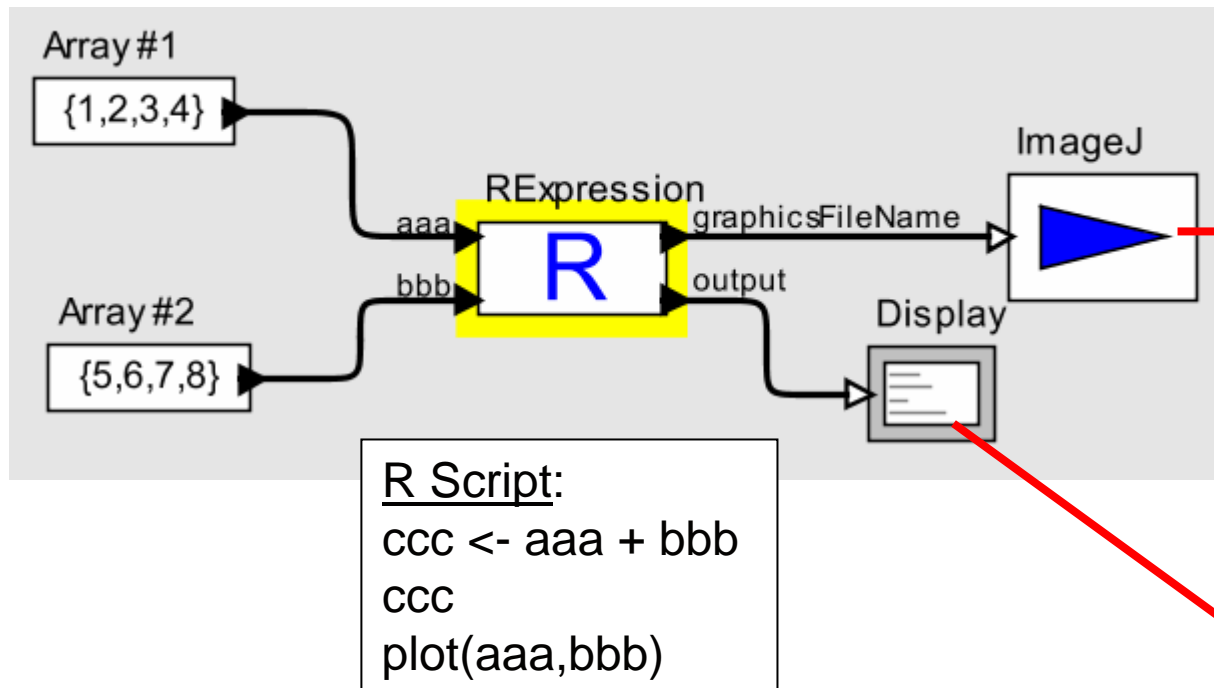
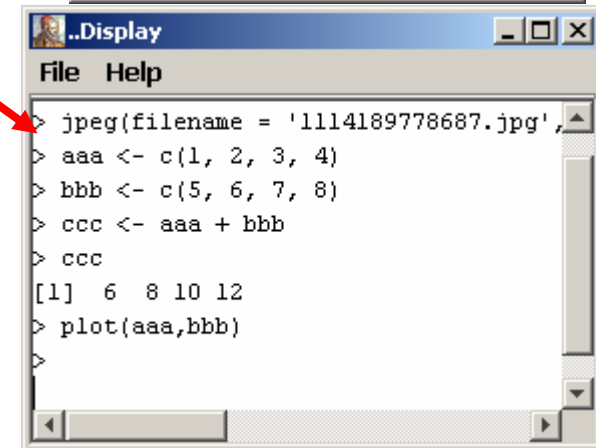
class: org.ecoinformatics.seek.R.RExpression

semanticType000: urn:lsid:localhost:onto:1:1#MathOperationActor

firingsPerIteration: 1

Commit Add Remove Restore Defaults Preferences Help Cancel

# Arrays and Graphical Output

```
> jpeg(filename = '1114189778687.jpg',
> aaa <- c(1, 2, 3, 4)
> bbb <- c(5, 6, 7, 8)
> ccc <- aaa + bbb
> ccc
[1] 6 8 10 12
> plot(aaa,bbb)
>
```

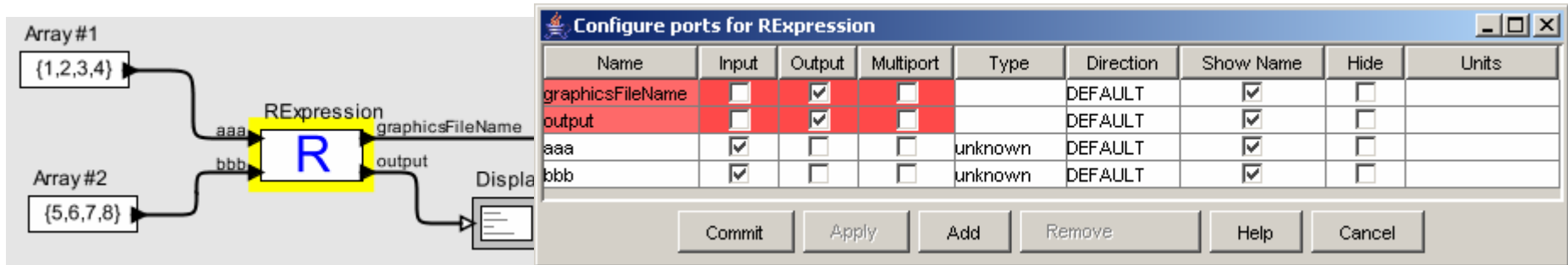
Adding ports automatically creates R objects with the port name [e.g. `aaa <- c(1,2,3,4)`]

Graphics automatically saved as images and sent to 'graphicsFileName' output port (as file name)

R text output automatically sent to 'output' port



# RExpression – Ports & Parameters



Edit parameters for RExpression

R function or script:

```
ccc <- aaa + bbb  
ccc  
plot(aaa,bbb)
```

R working directory:  
save or not: --no-save  
graphicsOutput: ☒  
Number of X pixels in image: 240  
Number of Y pixels in image: 240  
firingsPerIteration: 1

Buttons: Commit, Add, Remove, Restore Defaults, Preferences, Help, Cancel

Adding ports creates R objects from Kepler tokens

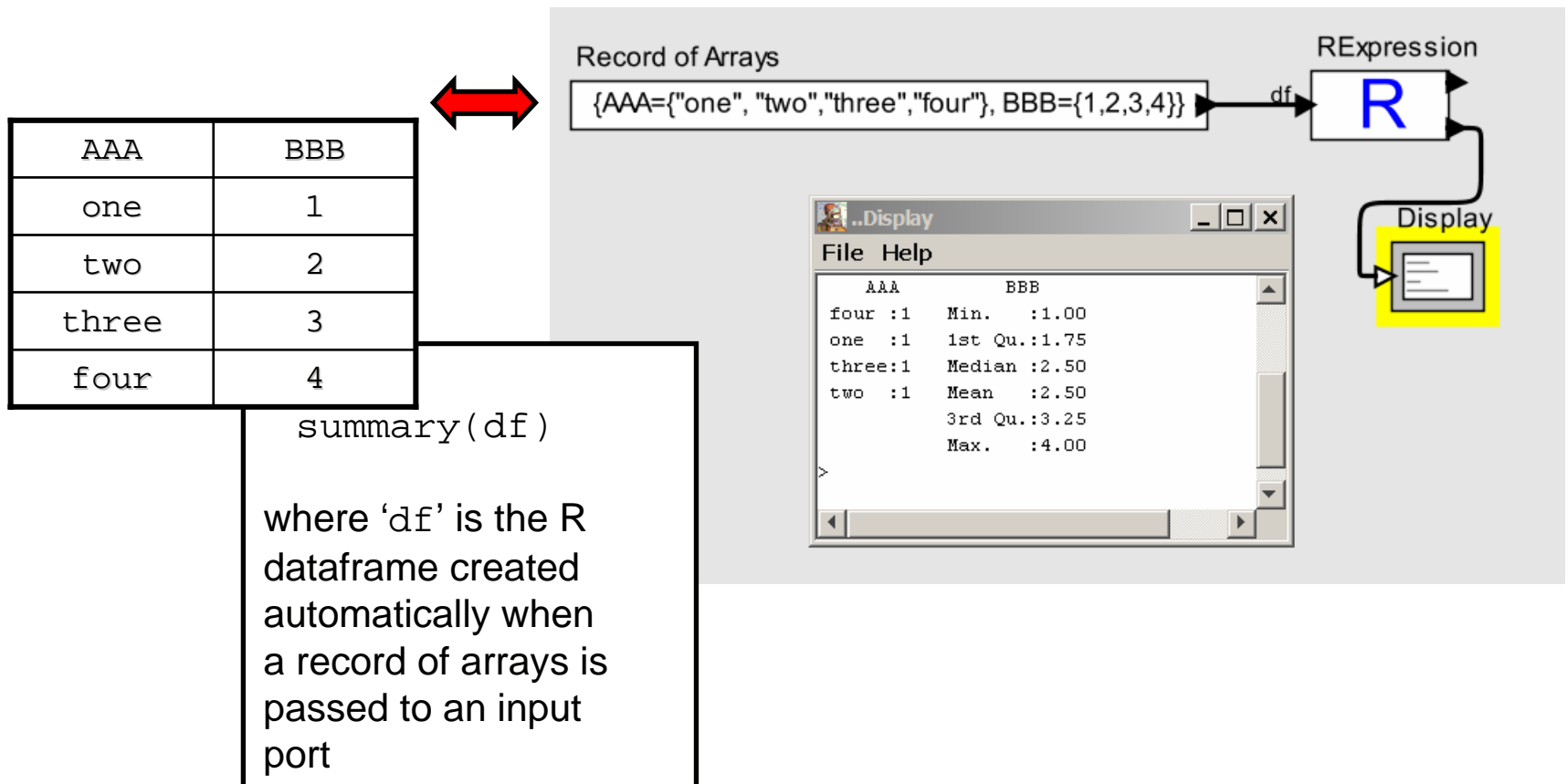
R script is a parameter of the RExpression actor which uses port names



# Array Records and Data Frames

Tables are represented as 'Data Frame' objects in 'R'

A Ptolemy 'Record of Arrays' can also represent a table





# RExpression Output Ports

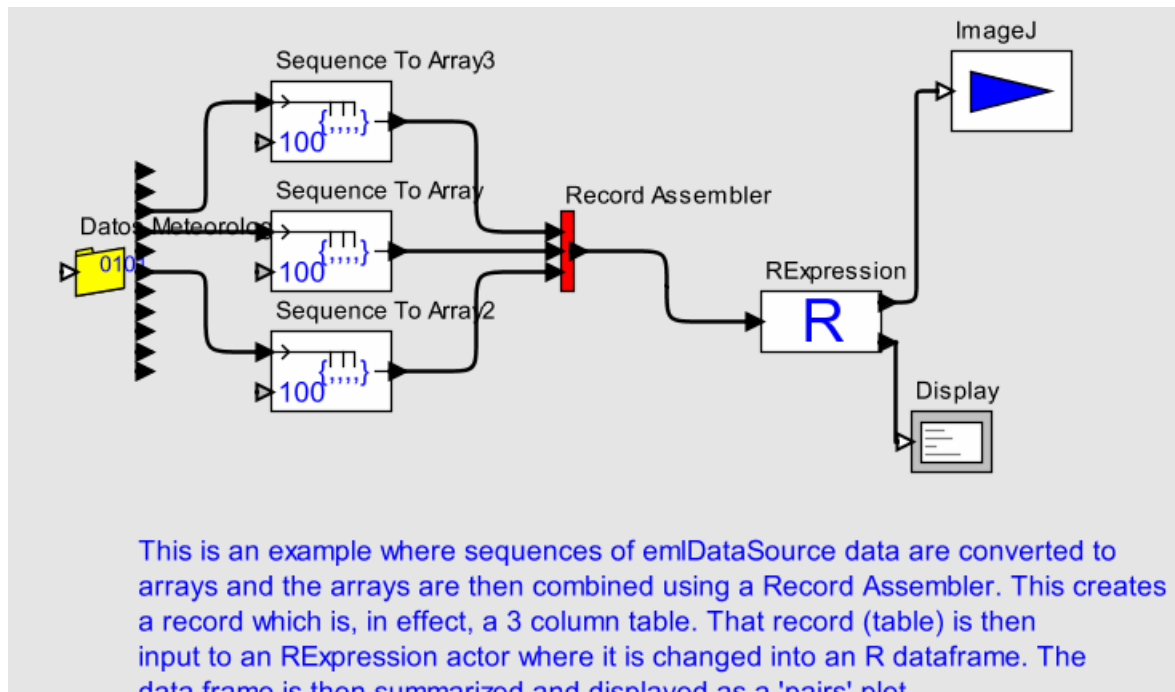
R vectors can also be assigned to output ports

- 1) Create a new output port with the “Configure Ports” dialog
- 2) Assign one of your arrays to the output port like so: `output1 <- input1`
- 3) Connect a “Display” actor to your new output port
- 4) Run the workflow

You should see your input1 array output on your output port.



# EML DataSource Sequence Inputs



EML DataSource actor provides table data from SEEK Ecogrid

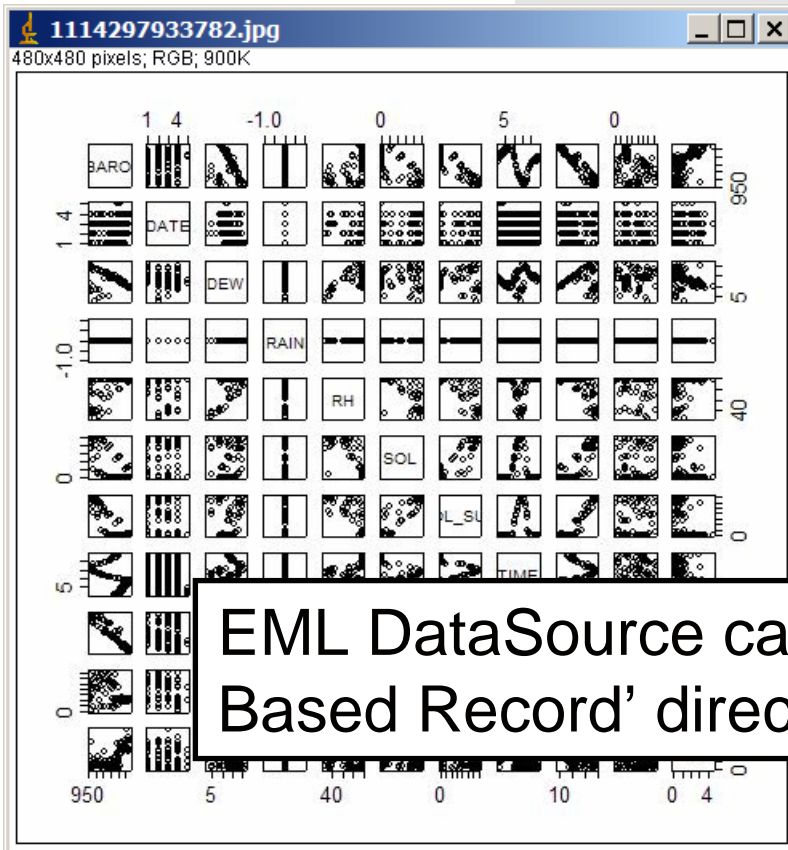
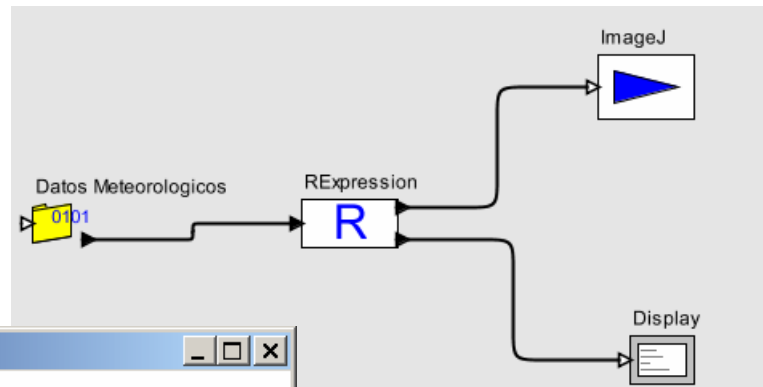
Column data from table can be supplied in various ways

Sequences of tokens from EML DataSource can be converted to arrays and then to a Record for input to RExpression



SEEK

# EML DataSource as Column Record

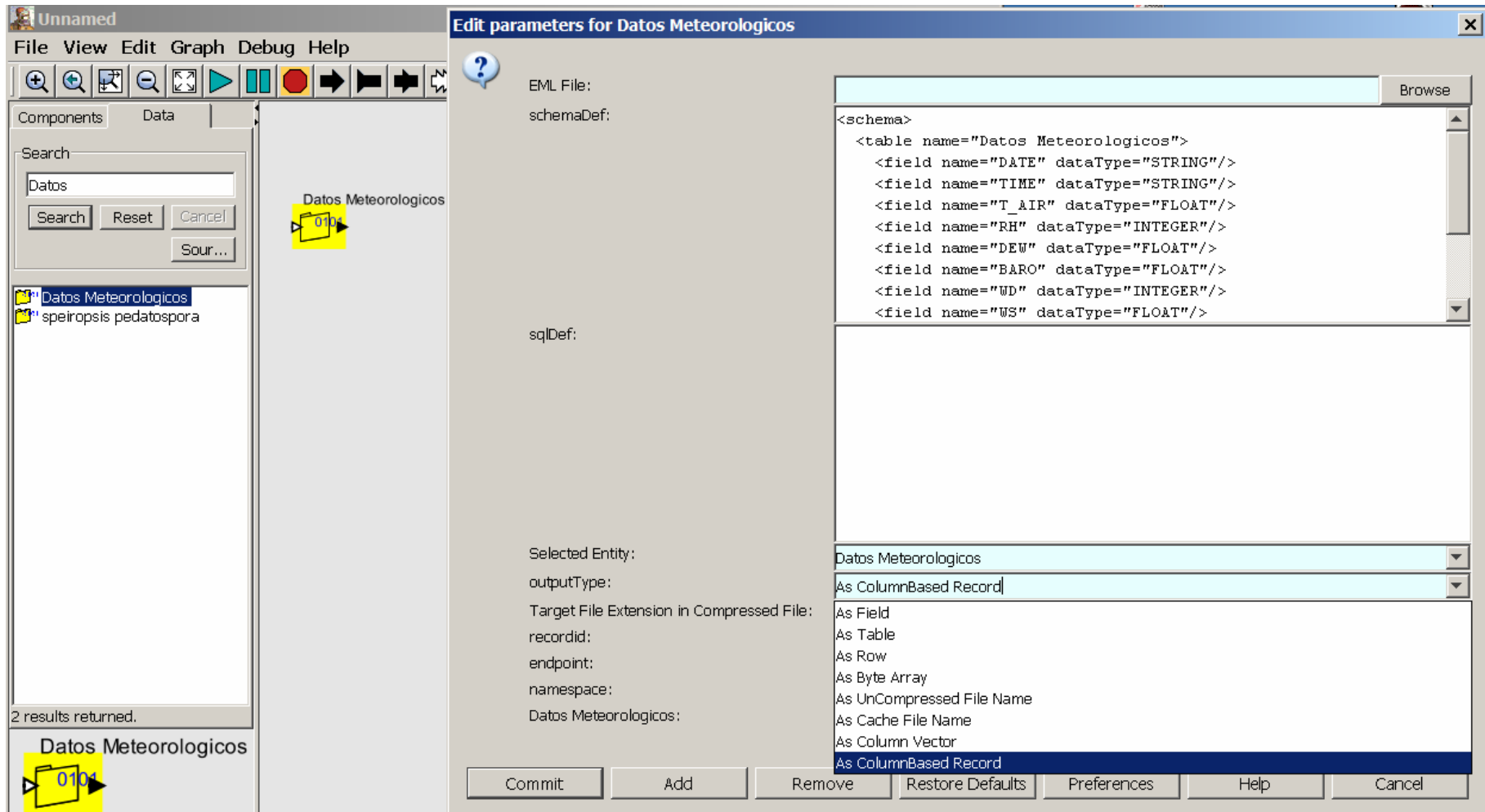


.eml\_Table\_as\_Record\_R.Display  
File Help

SOL	SOL_SUM	TIME	T_AIR
Min. : 0.0	Min. : 0	00:00 : 5	Min. : 8.90
1st Qu.: 0.0	1st Qu.: 0	01:00 : 5	1st Qu.:12.20
Median : 0.0	Median : 360	02:00 : 5	Median :15.15
Mean :258.0	Mean : 930382	03:00 : 5	Mean :16.06
3rd Qu.:564.5	3rd Qu.:1984590	04:00 : 4	3rd Qu.:20.15
Max. :982.0	Max. :3558000	05:00 : 4	Max. :24.40
		(Other):72	
WD	WS		
Min. : 2.00	Min. :0.000		
1st Qu.: 96.75	1st Qu.:0.300		
Median :113.50	Median :1.000		
Mean :157.43	Mean :1.335		
3rd Qu.:230.25	3rd Qu.:2.300		

EML DataSource can be configured to create a “Column Based Record” directly for input to RExpression

# Configuring a DataSource



The screenshot displays the SEEK application interface, which is used for configuring data sources. The main window is titled "Unnamed" and features a menu bar (File, View, Edit, Graph, Debug, Help) and a toolbar with various icons. The left sidebar shows a "Components" pane with a search bar and a list of components, including "Datos Meteorologicos" and "speiropsis pedatospora". The "Datos Meteorologicos" component is highlighted, and a yellow box with the number "010" is placed over it. The right pane is titled "Edit parameters for Datos Meteorologicos" and contains several configuration fields and a schema editor.

**Edit parameters for Datos Meteorologicos**

EML File:  Browse

schemaDef:

```
<schema>
  <table name="Datos Meteorologicos">
    <field name="DATE" dataType="STRING"/>
    <field name="TIME" dataType="STRING"/>
    <field name="T_AIR" dataType="FLOAT"/>
    <field name="RH" dataType="INTEGER"/>
    <field name="DEW" dataType="FLOAT"/>
    <field name="BARO" dataType="FLOAT"/>
    <field name="WD" dataType="INTEGER"/>
    <field name="WS" dataType="FLOAT"/>
  </table>
</schema>
```

sqlDef:

Selected Entity: Datos Meteorologicos

outputType: As ColumnBased Record

Target File Extension in Compressed File:

recordid:

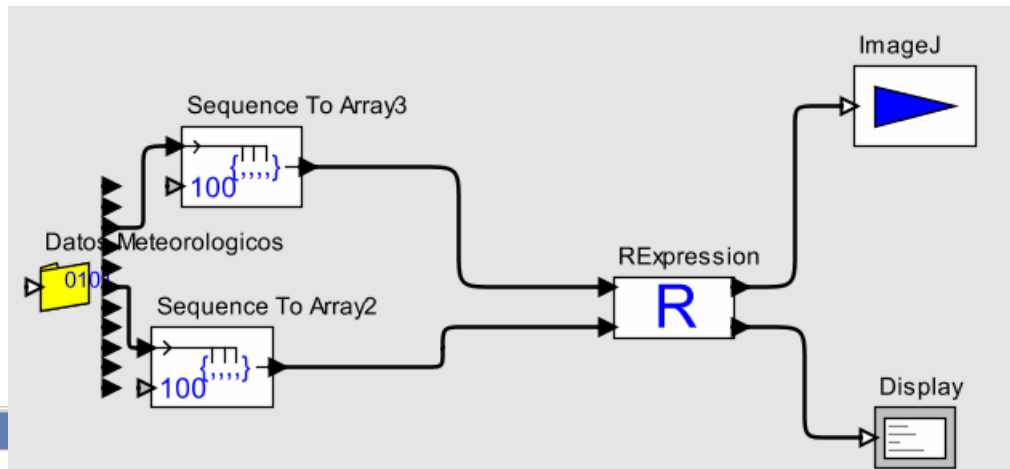
endpoint:

namespace: Datos Meteorologicos:

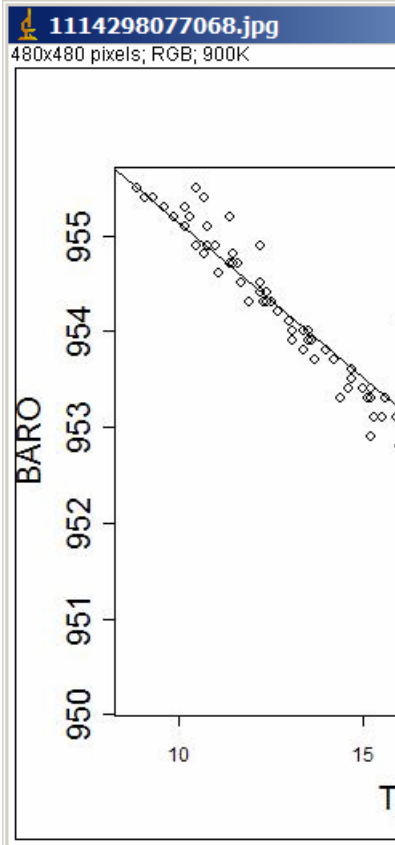
As Field  
As Table  
As Row  
As Byte Array  
As UnCompressed File Name  
As Cache File Name  
As Column Vector  
As ColumnBased Record

Commit Add Remove Restore Defaults Preferences Help Cancel

# R Regression Analysis Example



This is an example of how one can carry out a simple linear regression analysis using R and add the regression line to a scatter plot.  
Dan Higgins - March 2005

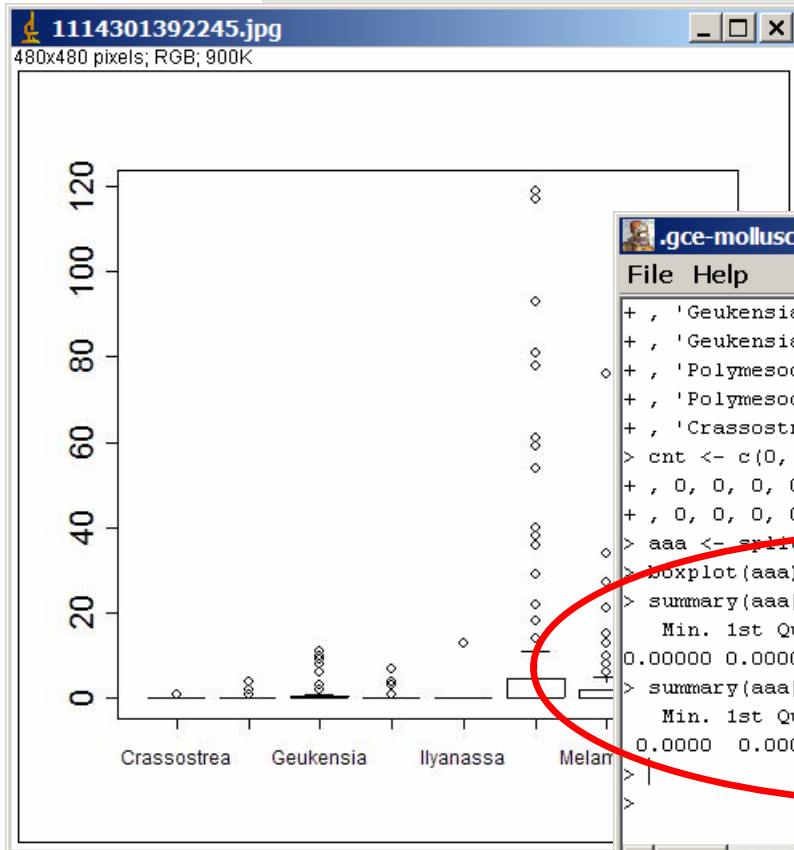


```
> T_AIR <- c(15.0, 13.4, 13.4, 12.4, 11.7, 11.4, 11.5, 11.5, 12.2, 17.4, 20.1,
> BARO <- c(953.4, 953.8, 954.0, 954.3, 954.5, 954.7, 954.8, 954.8, 954.9, 953.4, 954.8)
> res <- lm(BARO ~ T_AIR)
> res

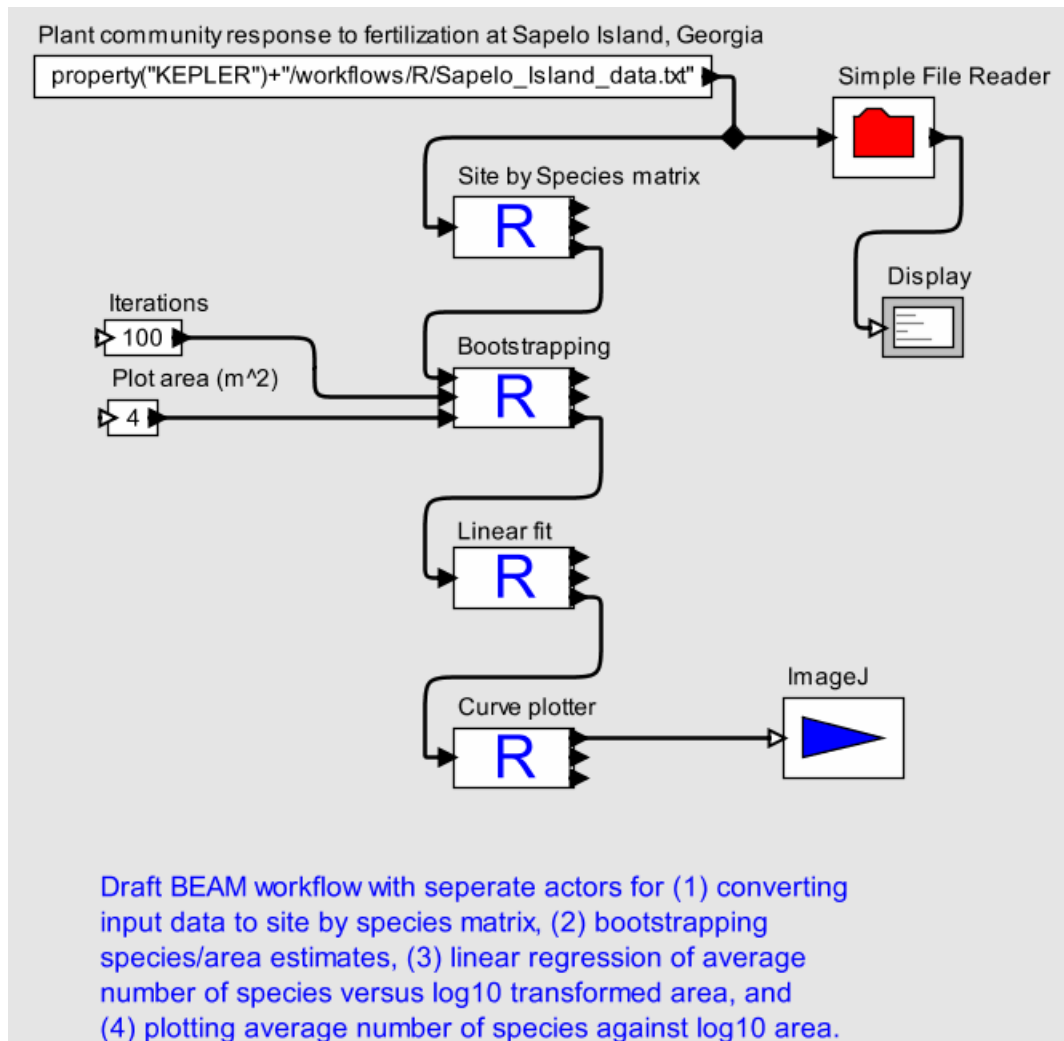
Call:
lm(formula = BARO ~ T_AIR)

Coefficients:
(Intercept)      T_AIR
  958.3772      -0.3244

> plot(T_AIR, BARO)
> abline(res)
>
```

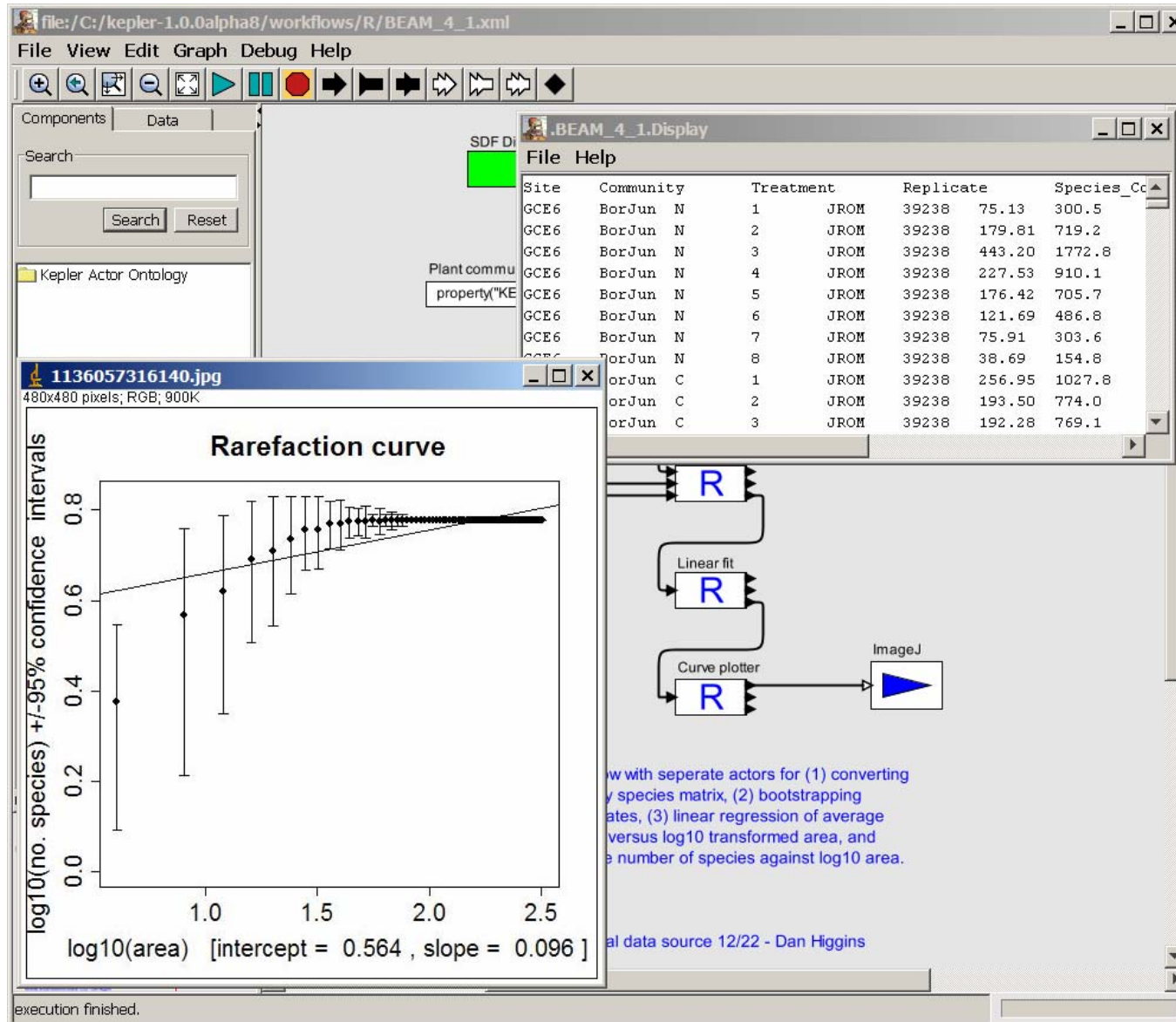
[illegible]

# Using Multiple R Actors





# Using Multiple R Actors - Result





# Interactive R in Kepler

file:/C:/ny/incoming/Kepler/kepler/workflows/R/InteractiveR.xml

File View Edit Graph Debug Help

Actors Data

Quick Search

Go

Directors  
Actors  
more libraries  
Utilities  
UserLibrary

SDF Director

SampleDelay

InteractiveShell

InteractiveExec

2+3

This is an example of interacting with an interactive subprocess ('R' in this specific case). The InteractiveShell actor is used to send a line to the InteractiveExec actor which loads a process and sends it the input result is then feedback for display. Note that the process remains between commands. ('R' must be installed locally for this workflow to operate properly.)

executing

R Graphics: Device 2 (ACTIVE)

Given: depth

Edgar Anderson's Iris Data

Sepal.Width

Petal.Length

Petal.Width

R Graphics: Device 2 (ACTIVE)

Frequency

Determinant

R Graphics: Device 2 (ACTIVE)

A Topographic Map of Maunga Whau

10 Meter Contour Spacing

Meters North

```
File Help
2+3
>>d <- outer(0:9, 0:9)
>
>>fr <- table(outer(d, d, "--"))
d <- outer(0:9, 0:9)
> fr <- table(outer(d, d, "--"))
>
>>plot(as.numeric(names(fr)), fr, type="h", xlab="Determinant", ylab="Frequency")
```



# Acknowledgements

This material is based upon work supported by:

The National Science Foundation under Grant Numbers 9980154, 9904777, 0131178, 9905838, 0129792, and 0225676.

Collaborators: NCEAS (UC Santa Barbara), University of New Mexico (Long Term Ecological Research Network Office), San Diego Supercomputer Center, University of Kansas (Center for Biodiversity Research), University of Vermont, University of North Carolina, Napier University, Arizona State University, UC Davis

The National Center for Ecological Analysis and Synthesis, a Center funded by NSF (Grant Number 0072909), the University of California, and the UC Santa Barbara campus.

The Andrew W. Mellon Foundation.

Kepler contributors: SEEK, Ptolemy II, SDM/SciDAC, GEON