ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-r



Ryan Smith
Peter Reutemann

January 8, 2020

©2012-2015



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato



Except where otherwise noted, this work is licensed under
http://creativecommons.org/licenses/by-sa/4.0/

Contents

1 Introduction **7**
1.1 Limitations 7

2 Setup **9**

3 Flow **11**
3.1 Actors 11
3.2 Examples 12
3.2.1 Standalone script 12
3.2.2 Generating data 14
3.2.3 Transforming data 16
3.2.3.1 Double matrix to double 16
3.2.3.2 Double matrix to double matrix 17
3.2.3.3 Double to double array 18
3.2.3.4 Spreadsheet to dataframe 20
3.2.4 Consuming data 22

4 Troubleshooting **25**
4.1 Windows 25
4.2 Tests 25

Bibliography **27**

List of Figures

3.1 Flow with standalone R script. 12
3.2 The standalone R script. 13
3.3 The generated plot. 13
3.4 Flow with RSource actor. 14
3.5 The data generating R script. 15
3.6 Plot of the random data generated by R. 15
3.7 Flow for calculating the determinant of a matrix. 16
3.8 Flow for transforming a double matrix. 17
3.9 Flow for generating spirals. 18
3.10 The R script for generating the spiral from the RTransformer actor. 19
3.11 The generated spirals plot. 19
3.12 Flow for generating a linear model from a spreadsheet. 20
3.13 Flow for plotting the residuals of a linear model. 21
3.14 The residuals of a linear model. 21
3.15 Flow with R script acting as sink. 22
3.16 The receiving R script. 23
3.17 The plot generated with R. 23

Chapter 1

Introduction

R is a language and environment for statistical computing and graphics. ADAMS-R provides an interface to R. It works by starting R as a server using Rserve[3], then communicating with Rserve through TCP. R code can be parsed and evaluated by Rserve through this connection and the result of any calculations can be returned.

1.1 Limitations

There are some limitations:

- Rserve does not provide any callback functionality so it cannot easily be used as a complete front-end for R;
- It should be possible to make plots within R and save them to the filesystem, but at this stage it is not possible to display R plots within the ADAMS system in any interactive way (other than as plain images) as Rserve lacks the callback ability of other interfaces such as JRI.
- The ability to run multiple simultaneous connections to Rserve is limited to **1** on Windows, according to `http://www.rforge.net/Rserve/doc.html#inst`: "Windows lacks important features that make the separation of namespaces possible, therefore Rserve for Windows works in cooperative mode only, that is only one connection at a time is allowed and all subsequent connections share the same namespace."

Chapter 2

Setup

1. The R software package is required, and is available here: `http://www.r-project.org/`.
2. Once R is installed, you need to install Rserve:
   - The easiest way to do this is to open R and type `install.packages("Rserve")`
   - Otherwise, if you are on a Unix-based system, you can type `R CMD INSTALL Rserve_1.7-0.tar.gz` on the command line.

   More detailed instructions can be found here: `http://www.rforge.net/Rserve/doc.html`.
3. Now you need to launch Rserve, there are two options for this:
   - The easiest way is to tell ADAMS the file path of R and Rserve using the preferences dialog in ADAMS, an example of a path to R on Mac OSX is: `~/Library/Frameworks/R.framework/Resources/bin/R64` and to Rserve is: `~/Library/R/2.15/library/Rserve/libs/x86_64/Rserve`. This allows ADAMS to start Rserve for you, whenever it needs to run.
   - Otherwise, you can start Rserve yourself by following the instructions here: `http://www.rforge.net/Rserve/doc.html`.

Chapter 3

Flow

3.1 Actors

The following flow actors are available:

- *RSource* – This can execute an R script and, like any other source actor, produces output (in the form of integers, doubles, strings, arrays of doubles, and matrices of doubles) to be passed through the flow.
- *RSink* – This sink takes input of the same types that RSource produces as output and executes a supplied R script, which can refer to the input data through the variable `X`, other flow variables can be referenced through `@{variable}`. Where `X` is used, RSink (and RTransformer) simply substitue that text for the name of an assigned variable in R, so to access an element of a matrix, for example, you would use `X[1][2]`, etc.
- *RTransformer* – This behaves much like a combination of RSource and RSink in that it takes input data, and produces output data. It also takes an R script and can access the input data just like RSink.
- *RStandalone* – This is basically just a way to execute an R script from within adams. It doesn't take any input or produce any output within the flow.

3.2 Examples

3.2.1 Standalone script

ADAMS allows you to simply run R scripts that neither have input nor output, but you can still use variables and placeholders defined within the ADAMS framework. The example flow[1] in Figure 3.1 uses the *RStandalone* actor to execute an R script (see Figure 3.2). This script uses an ADAMS variable for the filename of the generated plot.

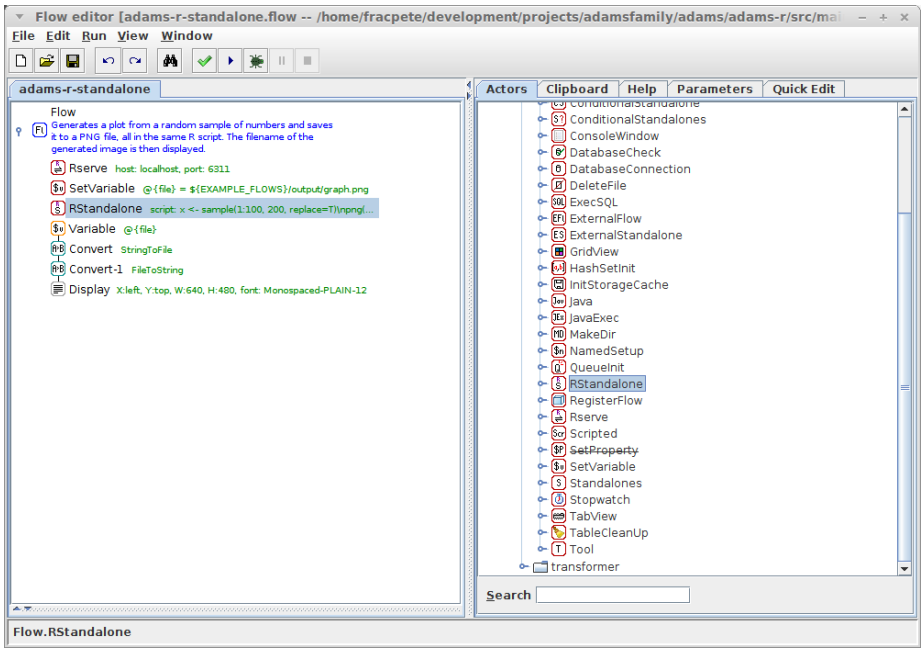


Figure 3.1: Flow with standalone R script.

[1] adams-r-standalone.flow

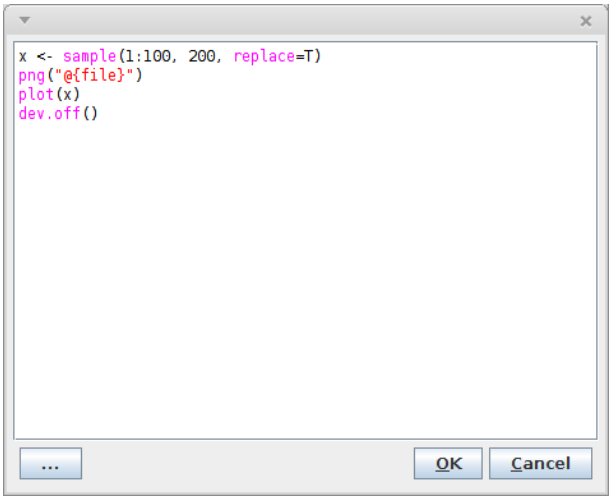


Figure 3.2: The standalone R script.

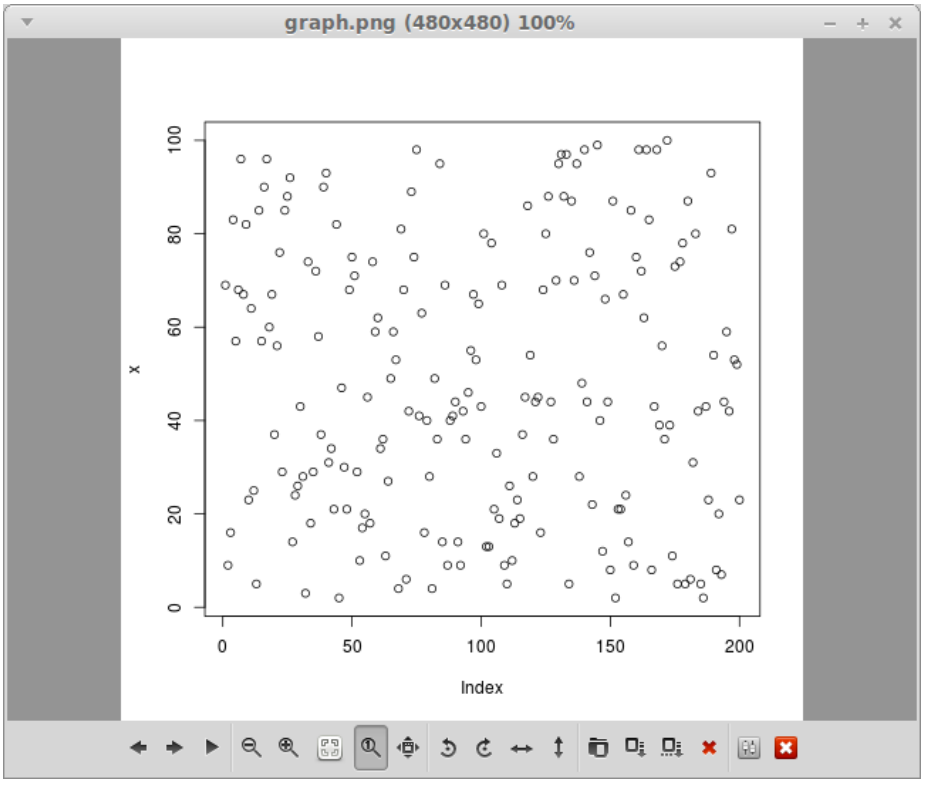


Figure 3.3: The generated plot.

3.2.2 Generating data

With the *RSource* actor you can use R to generate data and feed it into the flow like any other ADAMS source actor. The example flow[2] in Figure 3.4 generates an array of random numbers, transforms it with *log*2 and then uses ADAMS to plot the array data (see Figure 3.5).

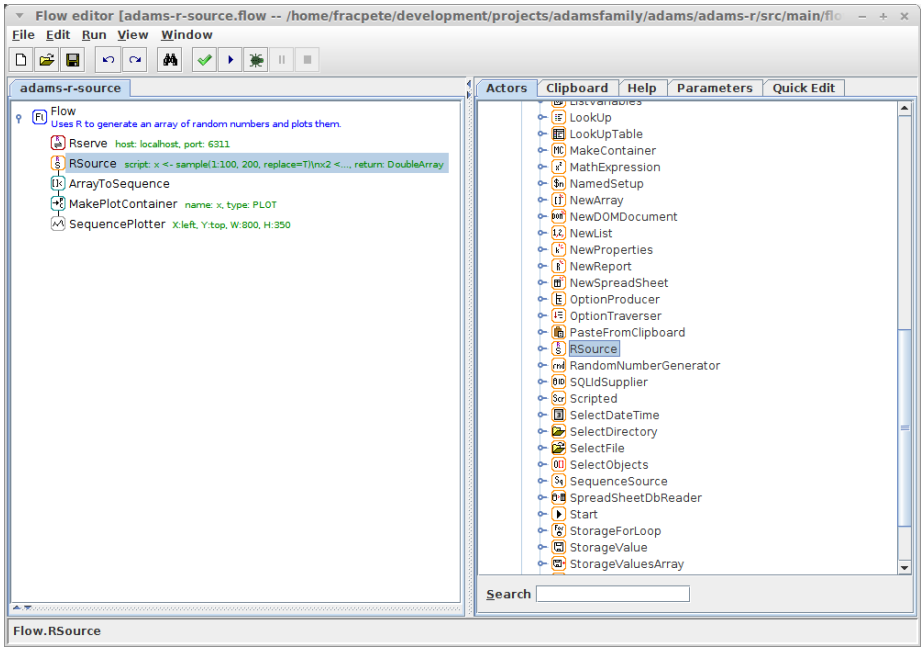


Figure 3.4: Flow with RSource actor.

[2] adams-r-source.flow

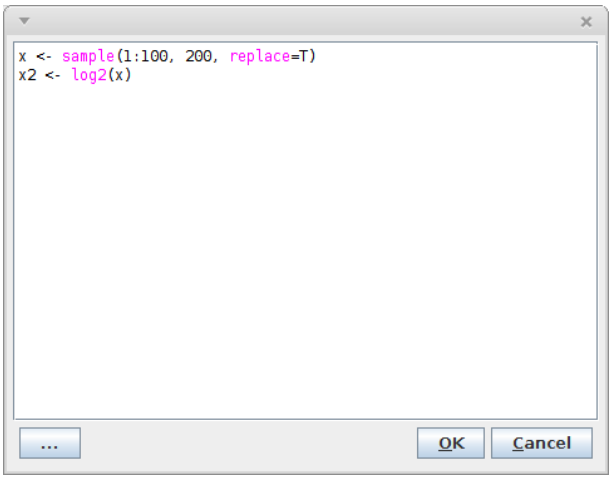


Figure 3.5: The data generating R script.

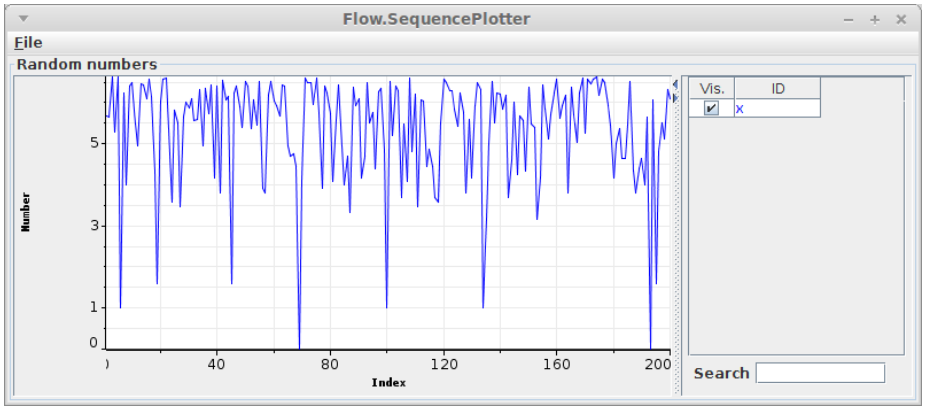


Figure 3.6: Plot of the random data generated by R.

3.2.3 Transforming data

Using the *RTransformer* actor, you can use R to easily transform data within a flow using R scripts. This allows you to use a plethora of R packages, all within the workflow environment.

3.2.3.1 Double matrix to double

R offers a lot of transformations and calculation around matrices. The example flow[3] turns a CSV string into a double matrix and calls R to calculate the determinant of the matrix (see Figure 3.7).

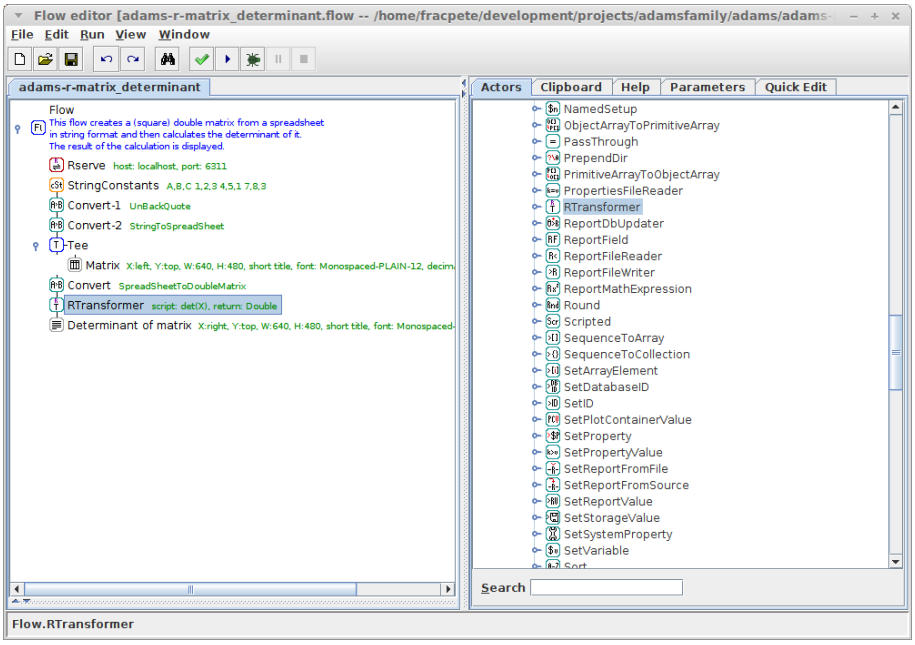


Figure 3.7: Flow for calculating the determinant of a matrix.

[3] adams-r-matrix_determinant.flow

3.2.3.2 Double matrix to double matrix

You can also turn matrices into matrices again, rather than just calculating a single value as in the previous example. The example flow[4] transforms the cells of the double matrix using *log*2. See Figure 3.8.

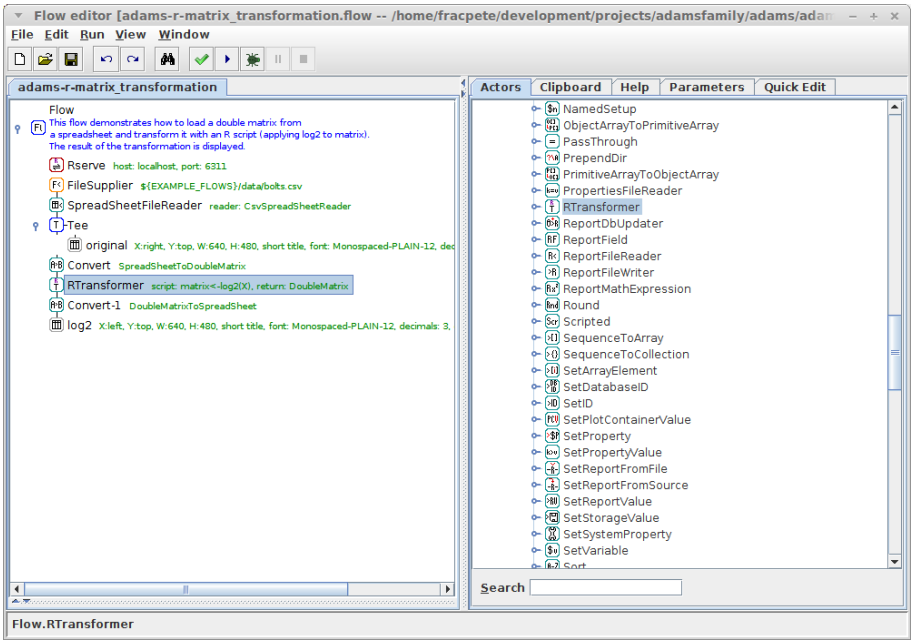


Figure 3.8: Flow for transforming a double matrix.

[4] adams-r-matrix_transformation.flow

3.2.3.3 Double to double array

This is an example of a flow that creates a pair of spirals[5]. It makes use of the RTransformer actor along with the Rserve actor to create an R server. The RTransformer makes use of a given x value and returns a pair of points, in the form of a double array, that represent the x and y values of the spiral. See Figures 3.9, 3.10 and 3.11.

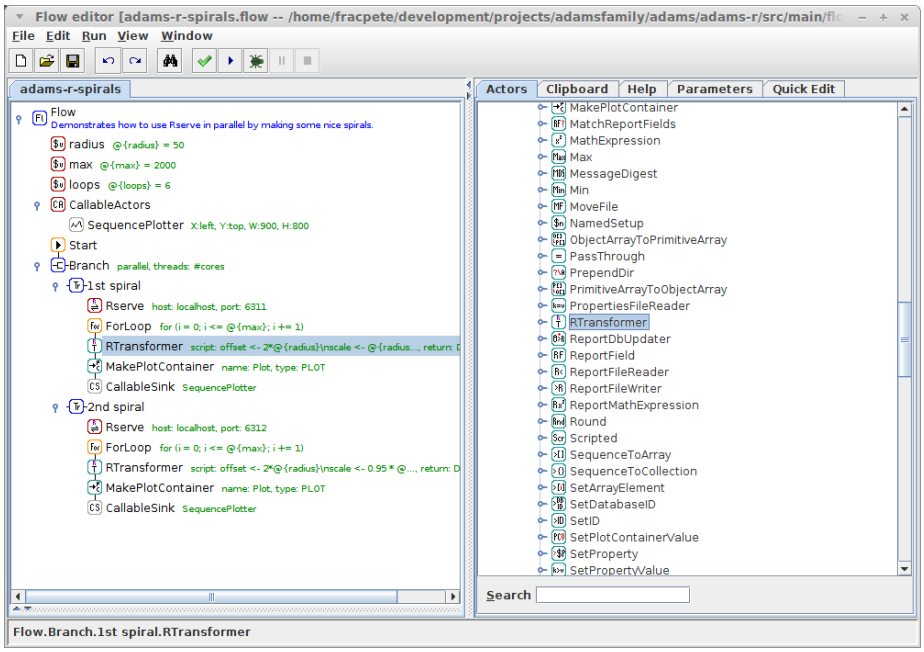


Figure 3.9: Flow for generating spirals.

[5] adams-r-spirals.flow

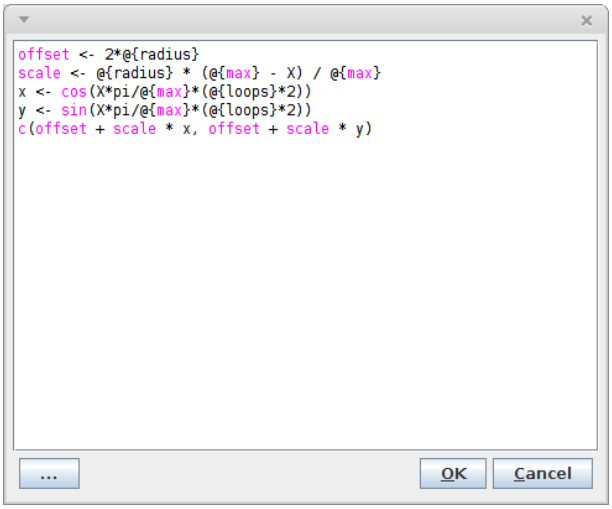


Figure 3.10: The R script for generating the spiral from the RTransformer actor.

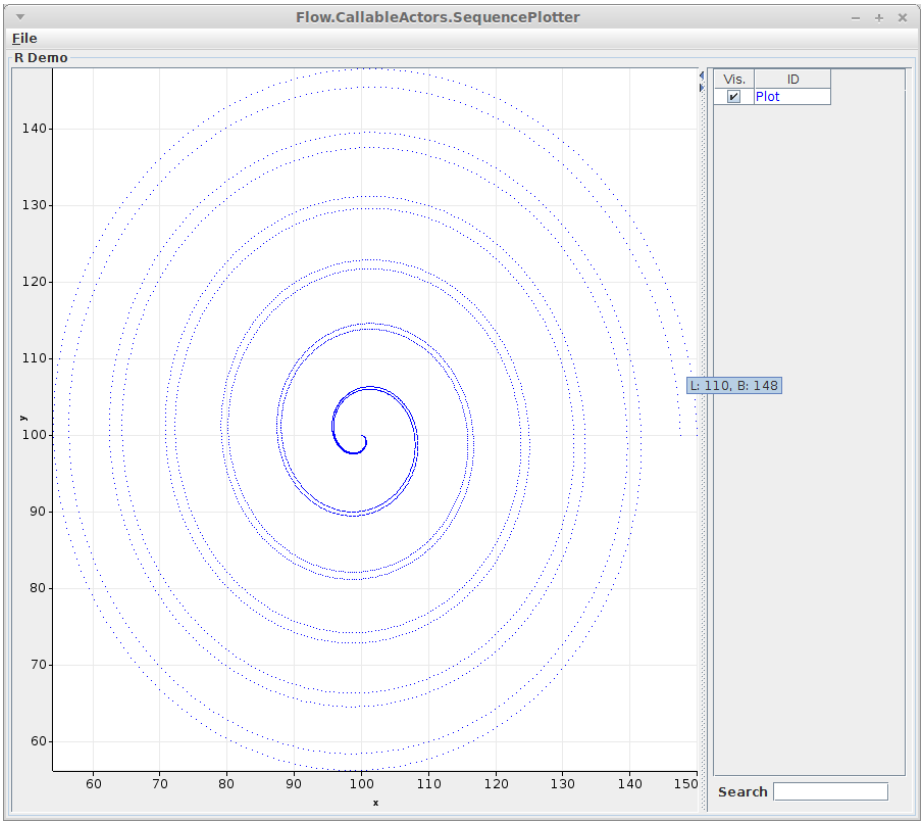


Figure 3.11: The generated spirals plot.

3.2.3.4 Spreadsheet to dataframe

Dataframes in R can be used to represent tables (or even nested structures). The example flow[6] loads a spreadsheet and generates a linear model using the `lm` command. The resulting dataframe is displayed as a spreadsheet again (see Figure 3.12). When generating a dataframe as output, you can limit the columns

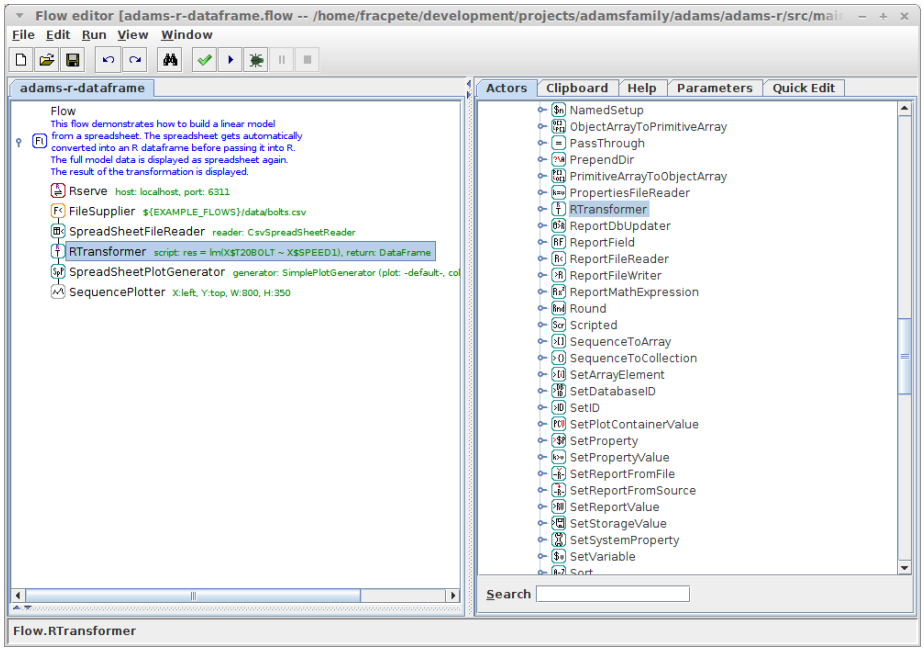


Figure 3.12: Flow for generating a linear model from a spreadsheet.

that should get returned in the spreadsheet. The example flow[7] in Figure 3.13 only retrieves the residuals from the linear model, which are displayed in Figure 3.14.

[6] adams-r-dataframe.flow
[7] adams-r-dataframe_columns.flow

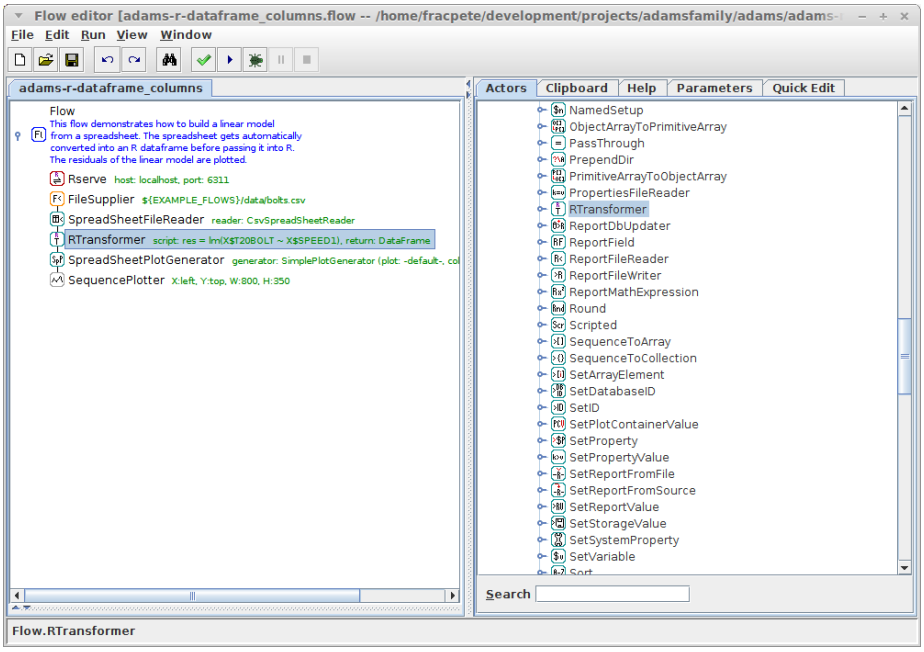


Figure 3.13: Flow for plotting the residuals of a linear model.

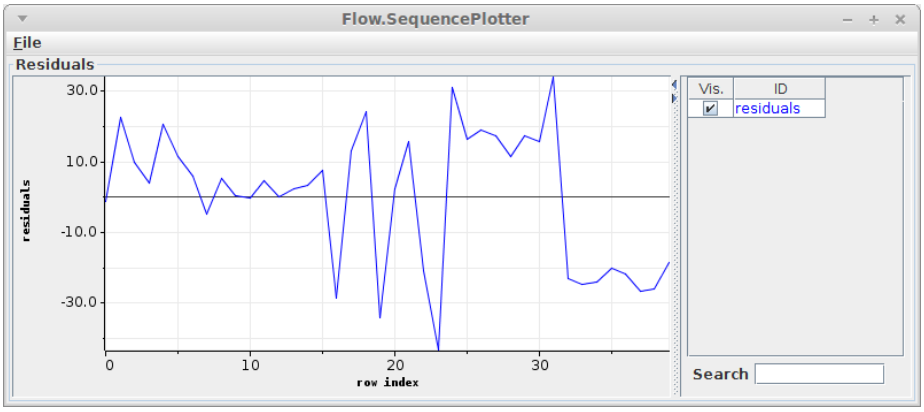


Figure 3.14: The residuals of a linear model.

3.2.4 Consuming data

Using the *RSink* actor, you can *consume* data generated with ADAMS with an R script. The example flow[8] in Figure 3.15 shows how to process an array of random doubles generated with ADAMS and generating a plot using R. Figure 3.16 shows the script used for the plot generation.

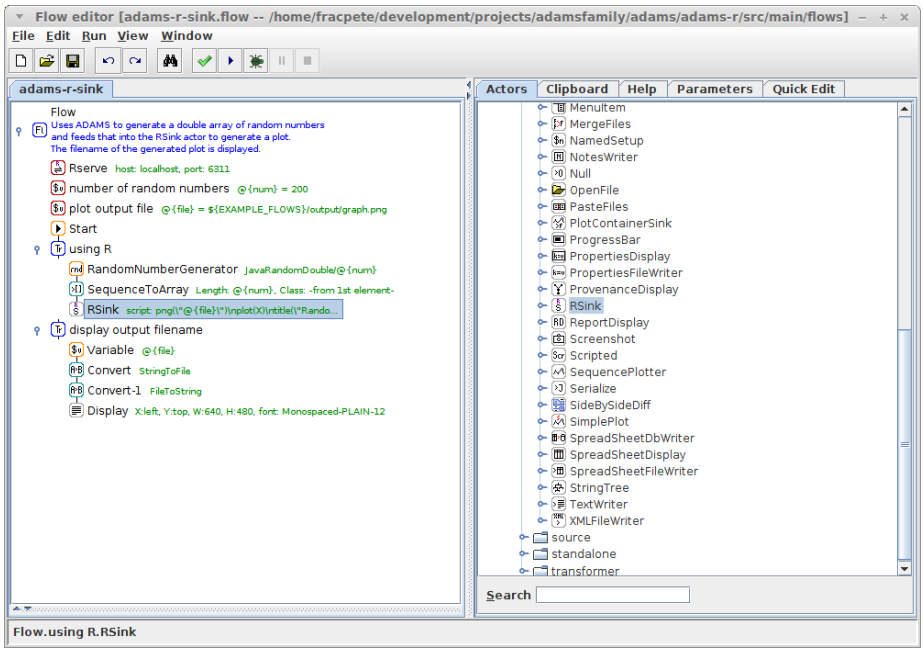


Figure 3.15: Flow with R script acting as sink.

[8] adams-r-sink.flow

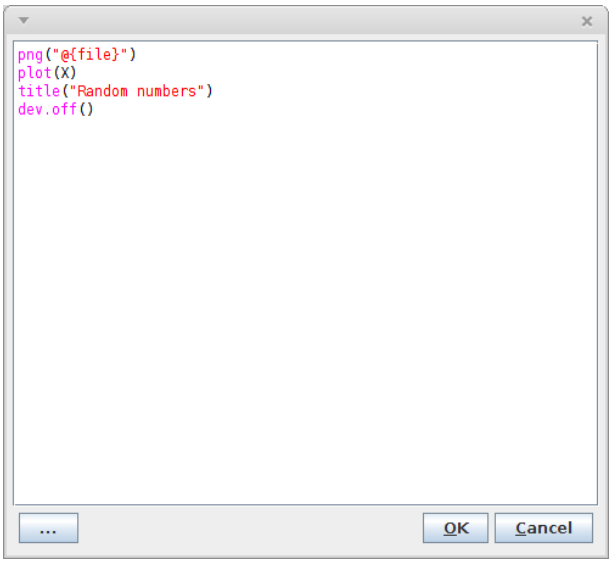


Figure 3.16: The receiving R script.

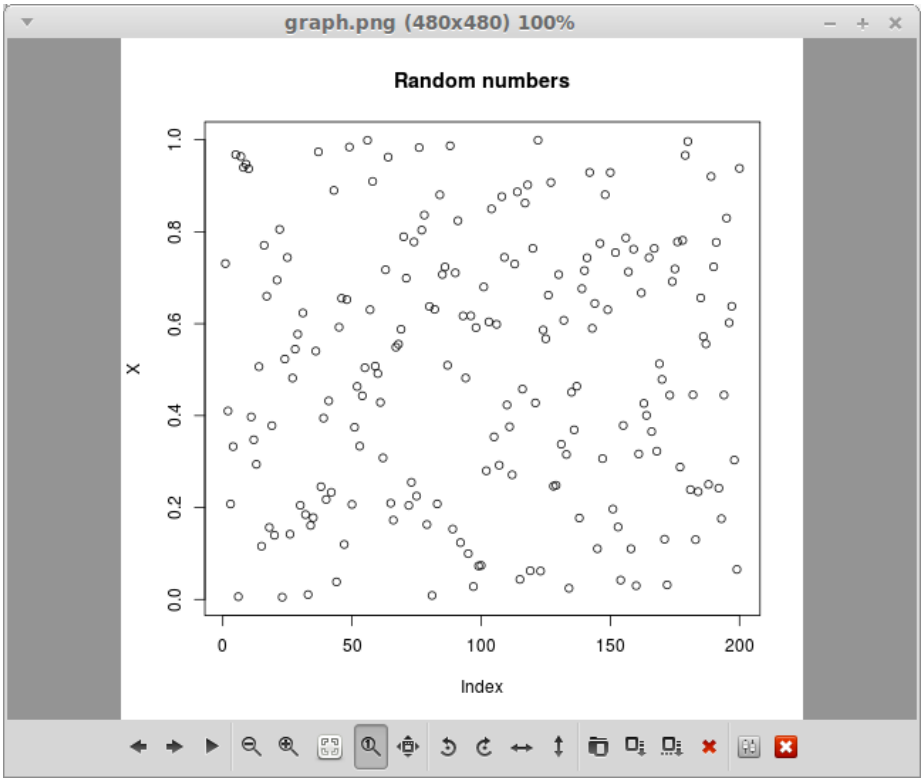


Figure 3.17: The plot generated with R.

Chapter 4

Troubleshooting

4.1 Windows

The flow hangs on execution – make sure that you only connect with one R actor to an Rserve server running on Windows (Linux/Unix/Mac allow an arbitrary number of connections). You can place *Rserve* standalone actors also inside *Trigger* control actors, specifying different ports.[1]

4.2 Tests

Junit tests of the flow actors can be disabled using the following command-line property:

```
-Dadams.test.flow.r.disabled=true
```

For instance, installing the *adams-r* module without running the R flow tests can be achieved with this command-line:

```
mvn clean install -Dadams.test.flow.r.disabled=true
```

[1] adams-r-spirals.flow

Bibliography

[1] *ADAMS* – Advanced Data mining and Machine learning System
`https://adams.cms.waikato.ac.nz/`

[2] *R Project* – The R Project for Statistical Computing
`http://www.r-project.org/`

[3] *RServe* – TCP/IP server allowing other programs to use facilities of R
`http://www.rforge.net/Rserve/`