Org-mode and julia: an introduction

G. Jay Kerns

February 27, 2013

Contents

One of the reasons for this document is that I wanted to make it easier to get acquainted with `julia`.

1 What you need to get started

This document assumes you have at least a passing familiarity with Org-mode and Emacs keybindings.

```
(load "/path/to/ob-julia.el")
(org-babel-julia-initiate-session "*julia*" nil)
```

Note: a lot of the code blocks below have the header argument `:eval no-export` which means that the code block can be evaluated interactively in this session by `C-c C-c` with point in the code block but will *not* be evaluated during export. The reason is that those blocks have settings which conflict with my current setup but would be useful for others going through this document.

1.1 Julia

- First install takes the longest, later updates not so bad.
- all the dependencies

1.2 Add-on packages

Based on The State of Statistics in Julia by John Myles White.

```
Pkg.add("DataFrames", "Distributions", "GLM", "MCMC", "Optim",
        "NHST", "Clustering")
```

```
Pkg.add("RDatasets")
```

1. Winston

   The most stable and fully featured of the `julia` graphics packages at the time of this writing appears to be the `Winston` package, among alternatives including `Gadfly`.

   ```
   Pkg.add("Winston")
   ```

The Winston package has lots of dependencies and many of them must be built from source (on Ubuntu).

2. Gadfly

   ```
   Pkg.add("Gadfly")
   ```

   - packages take a lot longer to load than R

1.3 Org-mode

This document assumes that you have at least a passing familiarity with org-mode such that you likely have something like the following already in your `.emacs`:

```
(require 'org)
```

Another handy setting to have is

```
(setq org-confirm-babel-evaluate nil)
```

In order to run this org file you will need to load `ob-julia.el` at some point. One way is to edit the following code block and then `C-c C-c` with point inside the block:

```
(load "/path/to/ob-julia.el")
(org-babel-julia-initiate-session "*julia*" nil)
```

The first command loads the `ob-julia.el` file and the second initiates a `julia` session in a buffer called `*julia*`. An alternative method is to put the following in your `.emacs` (these should go below the `(require 'org)` line):

```
(add-to-list 'load-path "/path/to/ob-julia.el")
(org-babel-do-load-languages
 'org-babel-load-languages
 '((emacs-lisp . t)
   (julia . t)))
```

The following lines (either here or in your `.emacs`) allow for inline image display in the Emacs buffer.

```
(add-hook 'org-babel-after-execute-hook 'org-display-inline-images)
(add-hook 'org-mode-hook 'org-display-inline-images)
```

If you'd like to do LaTeX export then put the following in your emacs.

```
(require 'ox-latex)
(require 'ox-beamer)
```

1.4 ESS - Emacs Speaks Statistics

The place to get the latest version of ESS is here.

```
(add-to-list 'load-path "/path/to/ESS/lisp")
(require 'ess-site)

(setq  inferior-julia-program-name "/path/to/julia-release-basic")
```

2 Fitting (generalized) linear models

```
using RDatasets, DataFrames, Distributions, GLM
trees = data("datasets", "trees");
treeslm = lm(:(Girth ~ Height + Volume), trees);
coef(treeslm)
coeftable(treeslm)
```

```
Warning: New definition show(Any,LmMod) is ambiguous with show(IO,ANY) at show.jl:6.
         Make sure show(IO,LmMod) is defined first.
Warning: New definition show(Any,GlmMod) is ambiguous with show(IO,ANY) at show.jl:6.
         Make sure show(IO,GlmMod) is defined first.
WARNING: strcat is deprecated, use string instead.
WARNING: qrd is deprecated, use qrfact instead.
3-element Float64 Array:
 10.8164
 -0.0454835
  0.19518
3x4 DataFrame:
          Estimate Std.Error  t value   Pr(>|t|)
[1,]       10.8164    1.9732  5.48165 7.44691e-6
[2,]    -0.0454835 0.0282621 -1.60935   0.118759
[3,]       0.19518 0.0109553  17.8161 8.2233e-17
```

3 Installation prerequisites

3.1 Org-mode

3.2 ESS

3.3 julia

4 Interactive session evaluation

This is about ESS.

5 Evaluation inside the Org buffer

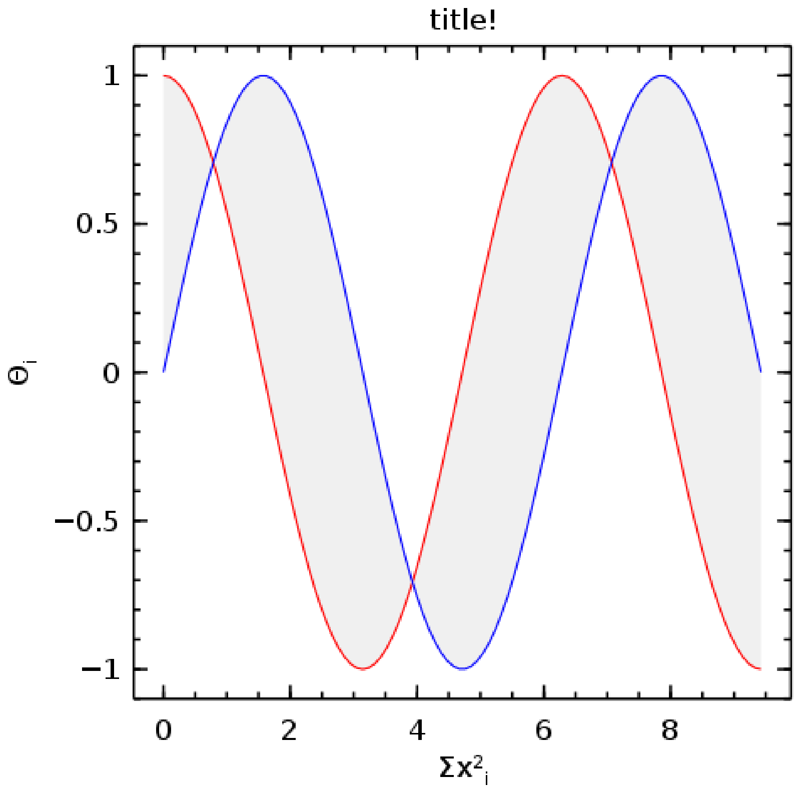
5.1 :results value

5.2 :results output

6 Graphics

6.1 Plotting with Winston

```
using Winston
x = linspace( 0, 3pi, 100 )
c = cos(x)
s = sin(x)
p = FramedPlot();
setattr(p, "title", "title!")
setattr(p, "xlabel", L"\Sigma x^2_i")
setattr(p, "ylabel", L"\Theta_i")
add(p, FillBetween(x, c, x, s) )
add(p, Curve(x, c, "color", "red") )
add(p, Curve(x, s, "color", "blue") )
file(p, "example1.png")
```



6.2 Plotting with Gadfly

```
using RDatasets
using Gadfly
using Compose
iris = data("datasets", "iris")
p = plot(iris, {:x => "Sepal.Length", :y => "Sepal.Width"}, Geom.point);
SVG("iris_plot.svg", 6inch, 4inch)
```

7 Exporting to other formats

7.1 LaTeX

7.2 HTML

7.3 Beamer

8 Other things

- empty lines in output for semicoloned lines
- need to start session first
- when :results value be careful because of readcsv
  - characters
  - 1x1 matrix