Consortium des Equipements
de Calcul Intensif
en Fédération Wallonie-Bruxelles

Introduction to Scripting and Interpreted Languages

damien.francois@uclouvain.be
Novembre 2023

Goal of this session:

"Advocate the use of scripting languages (interpreted languages) alongside Fortran and C++ and help you choose the most suitable for your needs"

Agenda

1. Interpreters vs compilers
2. Octave, R, Python, Julia
3. Graphical User Interfaces & Literate programming
4. Additional Packages/Libraries/Modules
5. What to do when it is too slow
6. Using several of them at the same time

Interpreters vs Compilers

- A **compiler** reads the whole (text) code and produces a separate "binary" file that can be executed by the CPU.

  C/C++, Fortran, Java, Go, Rust, Haskel, …

- An **interpreter** reads each line of code and executes it by calling the corresponding functionalities in its own code.

  Bash, Python, PHP, Javascript, Ruby, …

Interpreters vs Compilers

The ugly truth…

- Many interpreters will pre-compile the code
- Some compilers compile not to CPU-specific machine instructions but to bytecode that is interpreted
- The bytecode interpreters sometimes re-compile the bytecode just before execution (JIT compiling)
- Interpreters exist for C and C++
- Compilers exist for Python
- The interpreter can be compiled or himself interpreted

Interpreters vs Compilers (simplified)

Compilers

- can apply code-wise powerful optimization
- practically have no run-time overhead

→ Speed

Interpreters

- allow easy code introspection
- offer high-level language constructs and tools

→ Ease of use

Interpreted languages

- Easier to **learn**
  - Many implementation details hidden
  - Can try and test code portions rapidly and easily
- Easier to **exchange**/reuse
  - The scripts are cross-platform by nearly design
  - Often built-in package management
- Faster development
  - More **convenient programming** and shorter programs
    - Offers many simplifications and shortcuts – no need to micromanage memory
    - Built-in support for mundane tasks (handle files, dates, plots, NAs, NANs, …)
    - Often rich standard library
  - **Easier to debug** and profile
    - GUI

1.

Why those four?

Why those four?

- All very much used in **scientific applications**
  - R (S/SPlus): strong for statistics
  - Octave (Matlab): strong for engineering
  - Python Scipy/Numpy (Canopy,Anaconda): strong for data science
  - Julia: strong for computational sciences
- All **free** and free.
- All now **performance**-oriented
- Fun fact: Most started as wrappers for Fortran code!

Why those four?

By contrast,

- Ruby, Perl: smaller bioinformatics-only community
- Javascript, PHP, Bash, TCL, Lua: totally different goal
- Matlab, IDL, Mathematica: not free

QuadQuickstart

Start the interpreter to follow along:

```
ml spider Octave
ml spider SciPy-bundle
ml spider R
ml spider Julia
```

Assignment

```
octave:1> a=1, b=2
a =  1
b =  2
octave:2> a
a =  1
octave:3> b
b =  2
```

```
> a<-1; b<-2
> a
[1] 1
> b
[1] 2
> 1->a; b=2
```

```
>>> a, b = 1, 2
>>> a
1
>>> b
2
```

```
julia> a, b = 1, 2
(1, 2)

julia> a
1

julia> b
2
```

Operators

```
octave:11> c=[a-b, a*b, a/b, a^b]
c =
  -1.00000   2.00000   0.50000   1.00000
octave:12> c.*c
ans =
   1.00000   4.00000   0.25000   1.00000
octave:13> c*c'
ans =  6.2500
```

```
> c=c(a-b, a*b, a/b, a^b)
> c
[1] -1.0  2.0  0.5  1.0
> c*c
[1] 1.00 4.00 0.25 1.00
> c%*%c
     [,1]
[1,] 6.25
```

```
>>> import numpy as np
>>> c=np.array((a-b, a*b, a/b, a**b))
>>> c * c
array([1.  , 4.  , 0.25, 1.  ])
>>> c @ c
6.25
```

```
julia> using LinearAlgebra

julia> c=(a-b, a*b, a/b, a^b)
(-1, 2, 0.5, 1)

julia> c.*c
(1, 4, 0.25, 1)

julia> c⋅c # type \cdot<TAB>
6.25
```

Sequences

```
octave:15> b=1:9
b =

   1   2   3   4   5   6   7   8   9
```

```
> b <- 1:9
> b
[1] 1 2 3 4 5 6 7 8 9
```

```
>>> import numpy as np
>>> b = np.arange(1,10)
>>> b
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
julia> b = 1:9
1:9

julia> collect(b)
9-element Vector{Int64}:
 1
 2
 3
 [...]
```

Matrices

```
octave:16> a=reshape(1:9, 3, 3)
a =

   1   4   7
   2   5   8
   3   6   9
```

```
> a=array(1:9, dim=c(3,3))
> a
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
>>> a = np.arange(1,10).reshape(3,3)
>>> a
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
julia> a=reshape(1:9,3,3)
3×3 reshape(::UnitRange{Int64}, 3, 3)
with eltype Int64:
 1  4  7
 2  5  8
 3  6  9
```

Matrices

```
octave:3> zeros(3,2)
ans =

   0   0
   0   0
   0   0

octave:4> ones(2,3)
```

```
>>> np.zeros((3,2))
array([[0., 0.],
       [0., 0.],
       [0., 0.]])
>>> np.ones((3,2))
array([[1., 1.],
       [1., 1.],
       [1., 1.]])
```

```
> matrix(0,3,2)
     [,1] [,2]
[1,]    0    0
[2,]    0    0
[3,]    0    0
> matrix(1,2,3)
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    1    1    1
```

```
julia> zeros(3,2)
3×2 Matrix{Float64}:
 0.0  0.0
 0.0  0.0
 0.0  0.0

julia> ones(2,3)
2×3 Matrix{Float64}:
 1.0  1.0  1.0
 1.0  1.0  1.0
```

Slicing

```
octave:18> a(2,3)
ans =  8
octave:19> a(1,:)
ans =

   1   4   7

octave:20> a(:,1)
```

```
> a[2,3]
[1] 8
> a[1,]
[1] 1 4 7
> a[,1]
[1] 1 2 3
```

```
>>> a[2,1]
8
>>> a[1,:]
array([4, 5, 6])
>>> a[0,:]
array([1, 2, 3])
>>>
```

```
julia> a[2,3]
8

julia> a[1,:]
3-element Vector{Int64}:
 1
 4
 7
```

Slicing

```
octave:27> b(1:2:end)
ans =

   1   3   5   7   9

octave:28> b(2:2:end-1)
ans =

   2   4   6   8
```

```
>>> b[::2]
Array([1, 3, 5, 7, 9])
>>> b[1::2]
array([2, 4, 6, 8])
```

```
> b[seq(1,9,2)]
[1] 1 3 5 7 9
> b[seq(2,8,2)]
[1] 2 4 6 8
```

```
julia> collect(b[1:2:end])
5-element Vector{Int64}:
 1
 3
 5
 7
 9

julia> collect(b[2:2:end-1])
```

Slicing

```
octave:36> b>5
ans =

  0  0  0  0  0  1  1  1  1

octave:37> find(b>5)
ans =

   6   7   8   9
```

```
>>> b>5
array([False, False, False, False,
False,  True,  True,  True,  True])
>>> np.where(b>5)
(array([5, 6, 7, 8]),)
```

```
> b>5
[1] FALSE FALSE FALSE FALSE FALSE
TRUE  TRUE  TRUE  TRUE
> which(b>5)
[1] 6 7 8 9
```

```
julia> b.>5
9-element BitVector:
 0
 0
 [...]
 1
 1

julia> findall(b.>5)
```

Loops

```
octave:38> for i=1:5
> i*i
> end
ans =  1
ans =  4
ans =  9
ans =  16
ans =  25
```

```
> for(i in 1:5) {
+ print(i*i)
+ }
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
```

```
>>> for i in range(1,6):
...   print(i*i)
...
1
4
9
16
25
```


```
julia> for i=1:5
       println(i*i)
       end
1
4
9
16
25
```

Conditionals

```
octave:40> if 2>1 ;
> disp("OK")
> end
OK
```

```
> if (2>1) {
+ print('OK')
+ }
[1] "OK"
```

```
>>> if 2>1:
...   print("OK")
...
OK
```

```
julia> if 2>1 ;
       println("OK")
       end
OK
```

Strings

```
octave:18> n = str2double("10")
n =  10
octave:19> ischar(n)
ans = 0
octave:20> isfloat(n)
ans = 1
octave:21> printf('Number=%d\n', n)
Number=10
```

```
> n <- as.numeric("10")
> cat(sprintf("Number=%d\n", n))
Number=10
> library(glue)
> glue("Number={n}")
Number=10
```

```
>>> n = float("10")
>>> type(n)
<class 'float'>
>>> print("Number=%d" % n)
Number=10
>>> print("Number={:.0f}".format(n))
Number=10
>>> print(f"Number={n:.0f}")
Number=10
```

```
julia> n=parse(Int, "10")
10

julia> print("Number=$n")
Number=10
julia> using Printf

julia> @printf("Number=%.0f", n)
Number=10
```

Special numbers

```
octave:22> nan
ans =  NaN
octave:23> inf
ans =  Inf
```

```
> NaN
[1] NaN
> Inf
[1] Inf
> NA
[1] NA
> NULL
NULL
```

```
>>> np.nan
nan
>>> np.inf
Inf
>>> None
>>>
```

```
julia> NaN
NaN

julia> Inf
Inf

julia> nothing
```

More complete list

Hyperpolyglot

Numerical Analysis & Statistics: MATLAB, R, NumPy, Julia

a side-by-side reference sheet

sheet one: grammar and invocation | variables and expressions | arithmetic and logic | strings | regexes | dates and time | tuples | arrays | arithmetic sequences | 2d arrays | 3d arrays | dictionaries | functions | execution control | file handles | directories | processes and environment | libraries and namespaces | reflection | debugging

sheet two: tables | import and export | relational algebra | aggregation

vectors | matrices | sparse matrices | optimization | polynomials | descriptive statistics | distributions | linear regression | statistical tests | time series | fast fourier transform | clustering | images | sound

bar charts | scatter plots | line charts | surface charts | chart options

	matlab	r	numpy	julia
version used	*MATLAB 8.3* *Octave 3.8*	*3.1*	*Python 2.7* *NumPy 1.7* *SciPy 0.13* *Pandas 0.12* *Matplotlib 1.3*	*0.4*

show version	`$ matlab -nojvm -nodisplay -r 'exit'`
	`$ octave --version`
		`$ R --version`	`sys.version`
			`np.__version__`
			`sp.__version__`
			`mpl.__version__`
				`$ julia --version`

implicit prologue	*none*	`install.packages('ggplot2')`
		`library('ggplot2')`
			`import sys, os, re, math`
			`import numpy as np`
			`import scipy as sp`
			`import scipy.stats as stats`
			`import pandas as pd`
			`import matplotlib as mpl`
			`import matplotlib.pyplot as plt`	

grammar and invocation

	matlab	r	numpy	julia

interpreter	`$ cat >>foo.m`
	`1 + 1`
	`exit`
	`$ matlab -nojvm -nodisplay -r "run('foo.m')"`
	`$ octave foo.m`
		`$ cat >>foo.r`
		`1 + 1`
		`$ Rscript foo.r`
		`$ R -f foo.r`
			`$ cat >>foo.py`
			`print(1 + 1)`
			`$ python foo.py`
				`$ cat >>foo.jl`
				`println(1 + 1)`
				`$ julia foo.jl`

repl	`$ matlab -nojvm -nodisplay` `$ octave`	`$ R`	`$ python`	`$ julia`
command line program	`$ matlab -nojvm -nodisplay -r 'disp(1 + 1); exit'` `$ octave --silent --eval '1 + 1'`	`$ Rscript -e 'print("hi")'`	`python -c 'print("hi")'`	`$ julia -e 'println("hi")'`

block delimiters	`function end`
	`if elseif else end`
	`while end`
	`for end`
		`{ }`	*offside rule*	

3.

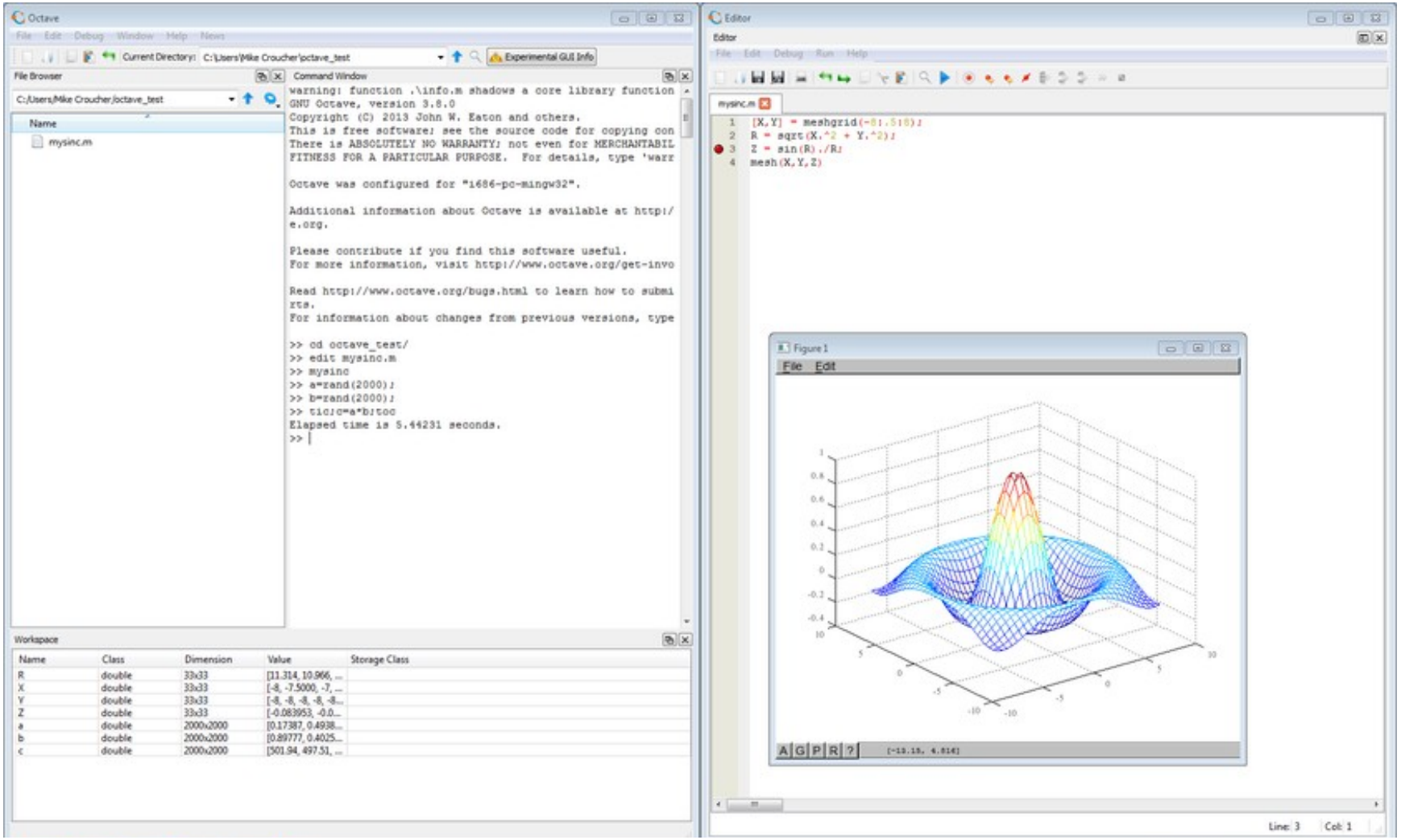
Graphical User Interfaces

Editing, debugging, accessing the doc, made easy

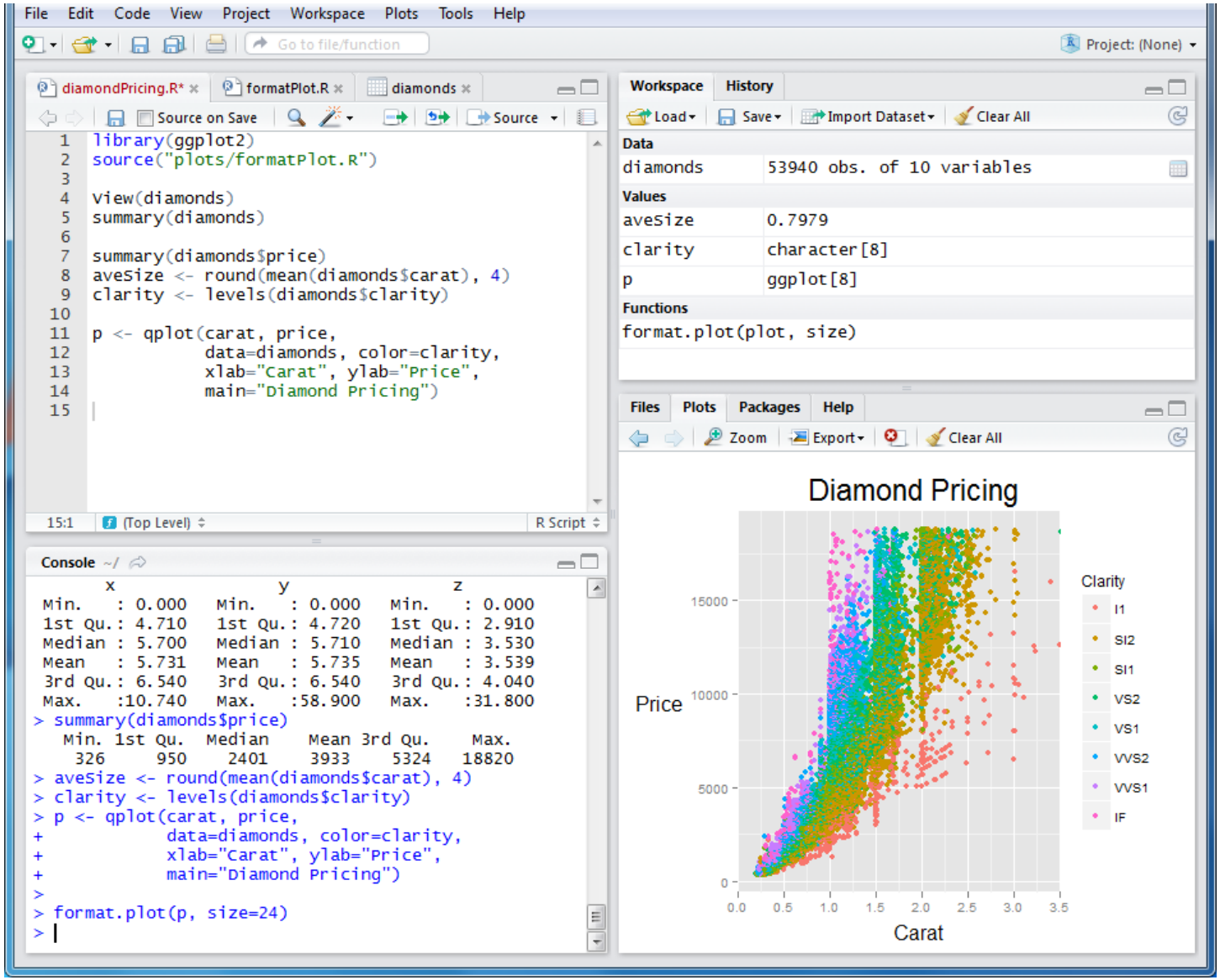
Literate programming

Authoring dynamic documents with code in them

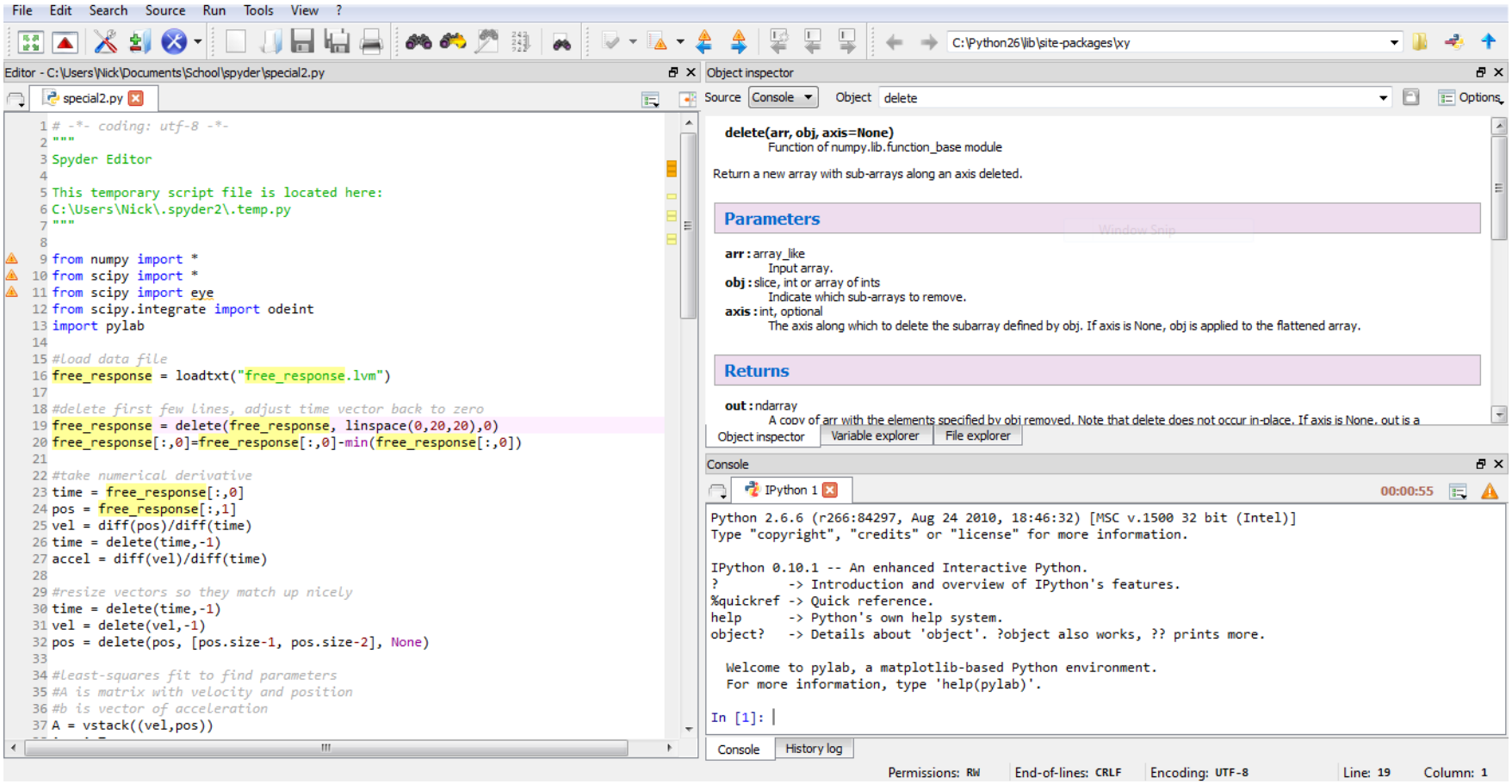
Octave


Octave
File Edit Debug Window Help News
Current Directory: C:\Users\Mike Croucher\octave_test
Experimental GUI Info
File Browser
C:/Users/Mike Croucher/octave_test
Name
mysinc.m
Command Window
warning: function .\info.m shadows a core library function
GNU Octave, version 3.8.0
Copyright (C) 2013 John W. Eaton and others.
This is free software; see the source code for copying con
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABIL
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warr
Octave was configured for "i686-pc-mingw32".
Additional information about Octave is available at http:/
e.org.
Please contribute if you find this software useful.
For more information, visit http://www.octave.org/get-invo
Read http://www.octave.org/bugs.html to learn how to submi
rts.
For information about changes from previous versions, type
>> cd octave_test/
>> edit mysinc.m
>> mysinc
>> a=rand(2000);
>> b=rand(2000);
>> tic;c=a*b;toc
Elapsed time is 5.44231 seconds.
>>
Workspace
Name Class Dimension Value Storage Class
R double 33x33 [11.314, 10.966, ...
X double 33x33 [-8, -7.5000, -7, ...
Y double 33x33 [-8, -8, -8, -8, -8...
Z double 33x33 [-0.083953, -0.0...
a double 2000x2000 [0.17387, 0.4938...
b double 2000x2000 [0.89777, 0.4025...
c double 2000x2000 [501.94, 497.51, ...
Editor
mysinc.m
1 [X,Y] = meshgrid(-8:.5:8);
2 R = sqrt(X.^2 + Y.^2);
3 Z = sin(R)./R;
4 mesh(X,Y,Z)
Figure 1
File Edit
A G P R ?
Line: 3
Col: 1

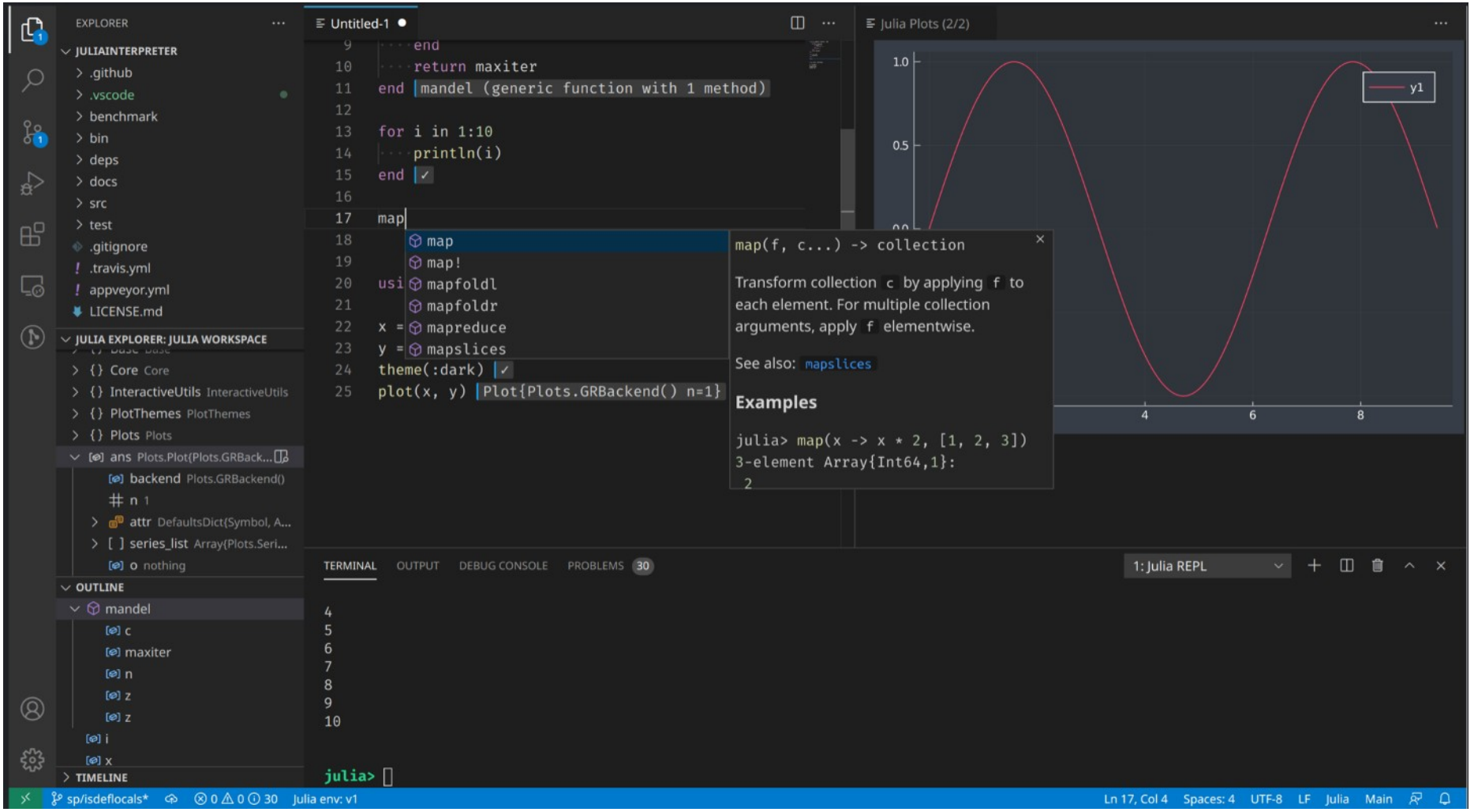
Rstudio (also for Python and Julia with some adaptations)



Spyder


File Edit Search Source Run Tools View ?
C:\Python26\lib\site-packages\xy
Editor - C:\Users\Nick\Documents\School\spyder\special2.py
special2.py
-*- coding: utf-8 -*-
"""
Spyder Editor
This temporary script file is located here:
C:\Users\Nick\.spyder2\.temp.py
"""
from numpy import *
from scipy import *
from scipy import eye
from scipy.integrate import odeint
import pylab
#load data file
free_response = loadtxt("free_response.lvm")
#delete first few lines, adjust time vector back to zero
free_response = delete(free_response, linspace(0,20,20),0)
free_response[:,0]=free_response[:,0]-min(free_response[:,0])
#take numerical derivative
time = free_response[:,0]
pos = free_response[:,1]
vel = diff(pos)/diff(time)
time = delete(time,-1)
accel = diff(vel)/diff(time)
#resize vectors so they match up nicely
time = delete(time,-1)
vel = delete(vel,-1)
pos = delete(pos, [pos.size-1, pos.size-2], None)
#least-squares fit to find parameters
#A is matrix with velocity and position
#b is vector of acceleration
A = vstack((vel,pos))
Object inspector
Source Console Object delete Options
delete(arr, obj, axis=None)
Function of numpy.lib.function_base module
Return a new array with sub-arrays along an axis deleted.
Parameters
arr : array_like
Input array.
obj : slice, int or array of ints
Indicate which sub-arrays to remove.
axis : int, optional
The axis along which to delete the subarray defined by obj. If axis is None, obj is applied to the flattened array.
Returns
out : ndarray
Object inspector Variable explorer File explorer
Console
IPython 1
00:00:55
Python 2.6.6 (r266:84297, Aug 24 2010, 18:46:32) [MSC v.1500 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.
IPython 0.10.1 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object'. ?object also works, ?? prints more.
Welcome to pylab, a matplotlib-based Python environment.
For more information, type 'help(pylab)'.
In [1]:
Console History log
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 19 Column: 1

Julia for Visual Studio Code (also for R & Python)



https://www.julia-vscode.org

RMarkdown and KnitR

chunks.Rmd

Knit HTML | Chunks

```
1  R Code Chunks
2  ==================================
3
4  With R Markdown, you can insert R code
   chunks including plots:
5
6  ```{r qplot, fig.width=4, fig.height=3,
   message=FALSE}
7  # quick summary and plot
8  library(ggplot2)
9  summary(cars)
10 qplot(speed, dist, data=cars) +
11       geom_smooth()
12 ```
13
```

RStudio: Preview HTML

Preview: ~/chunks.html | Save As | Publish

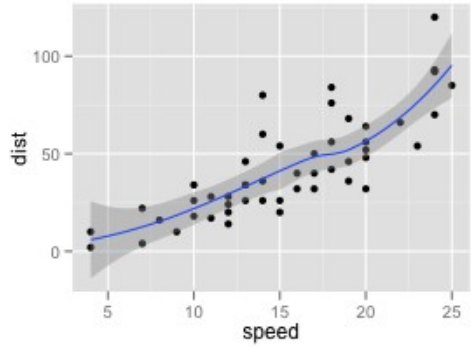
R Code Chunks

With R Markdown, you can insert R code chunks including plots:

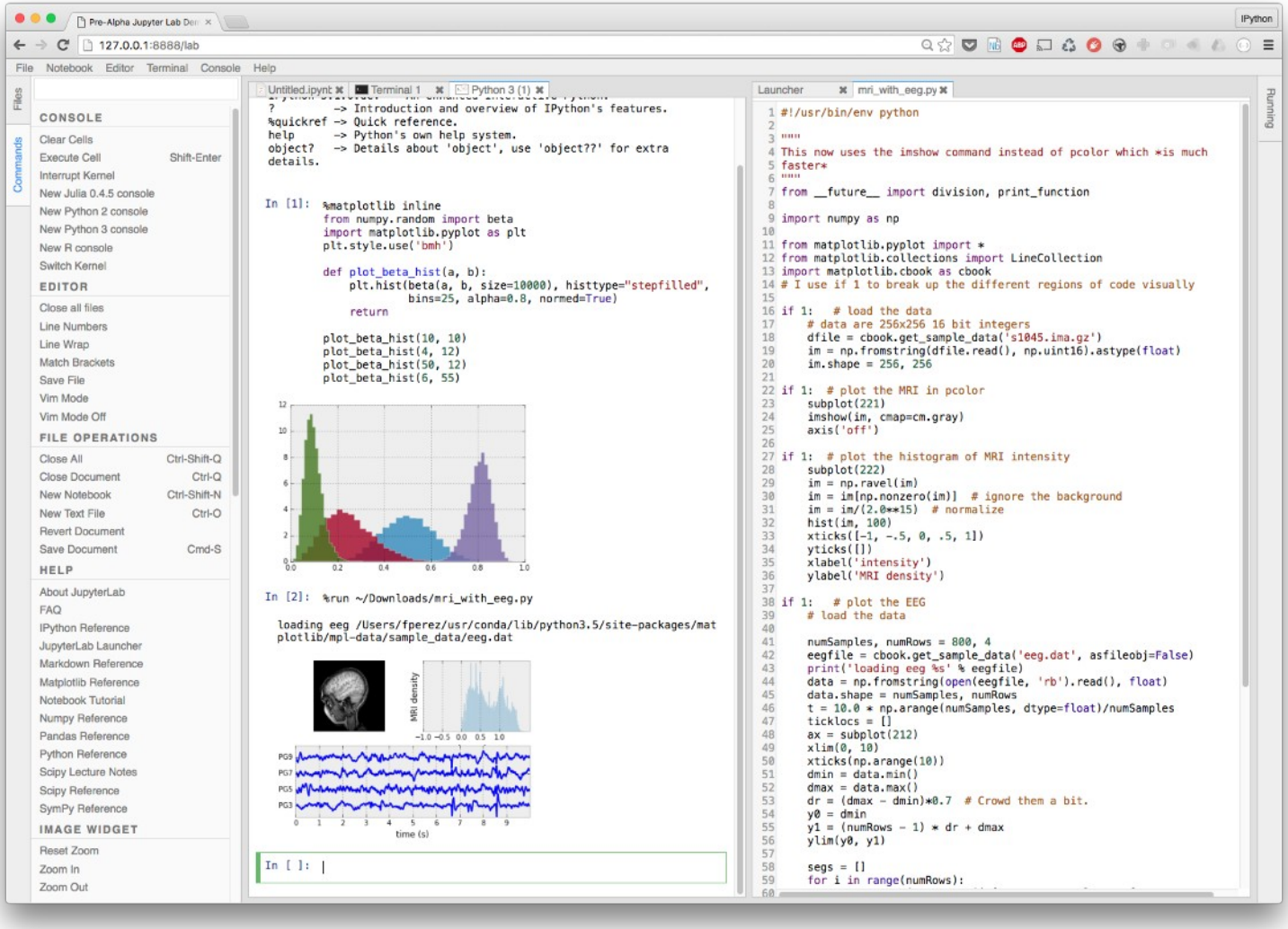
```
# quick summary and plot
library(ggplot2)
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2
##  1st Qu.:12.0   1st Qu.: 26
##  Median :15.0   Median : 36
##  Mean   :15.4   Mean   : 43
##  3rd Qu.:19.0   3rd Qu.: 56
##  Max.   :25.0   Max.   :120
```

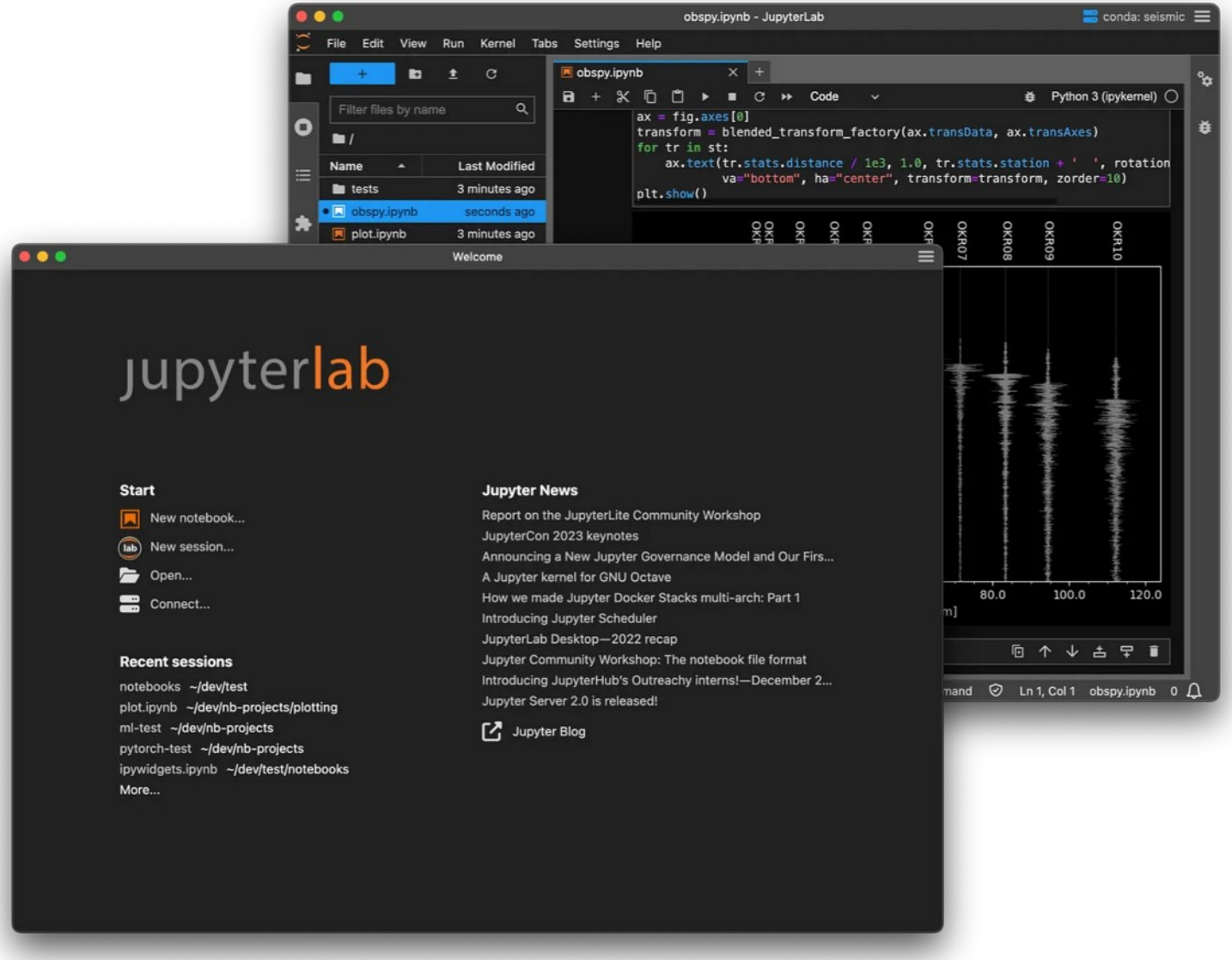
```
qplot(speed, dist, data = cars) + geom_smooth()
```



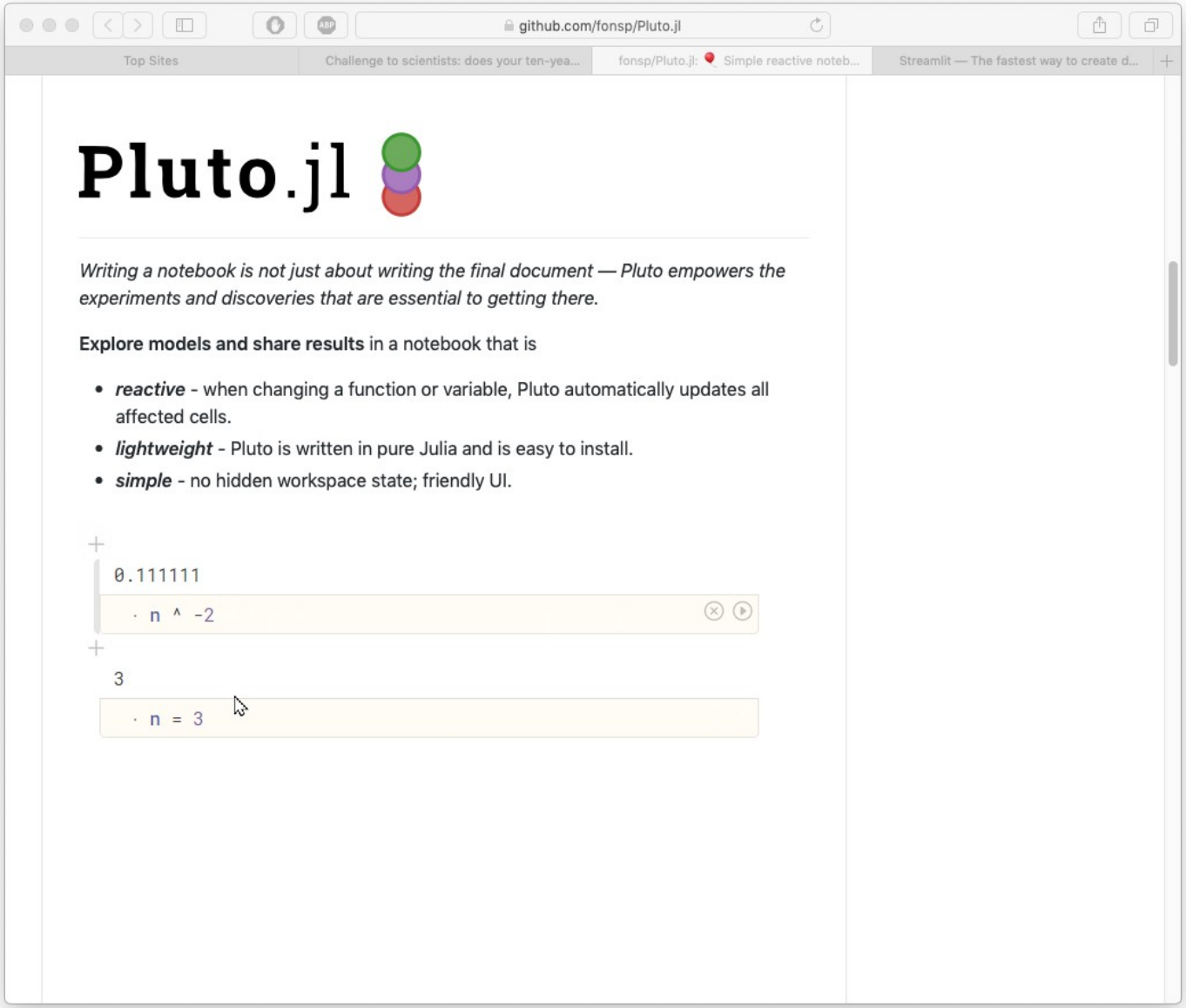
Jupyter notebooks (also for Julia & R)



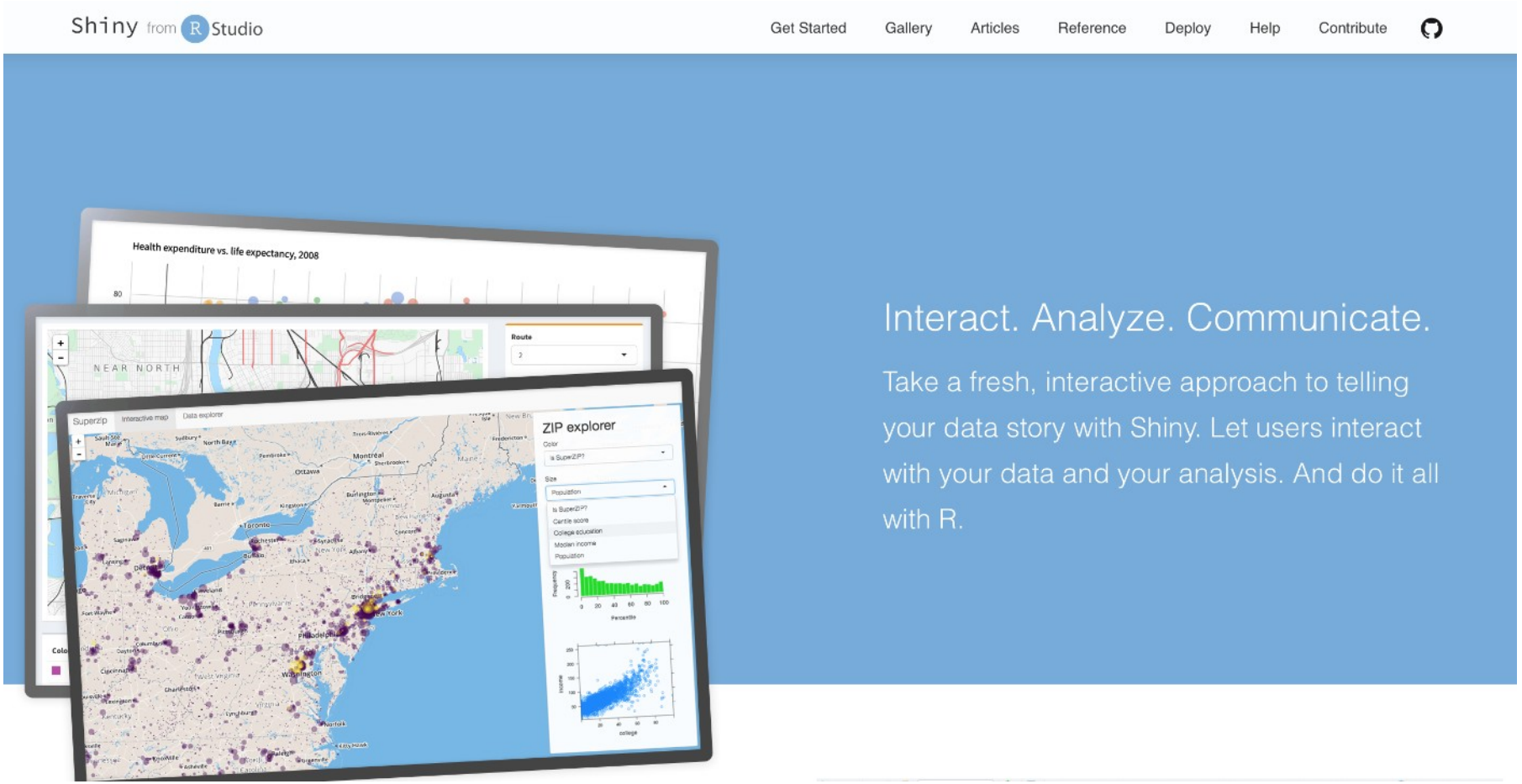
Jupyterlab desktop



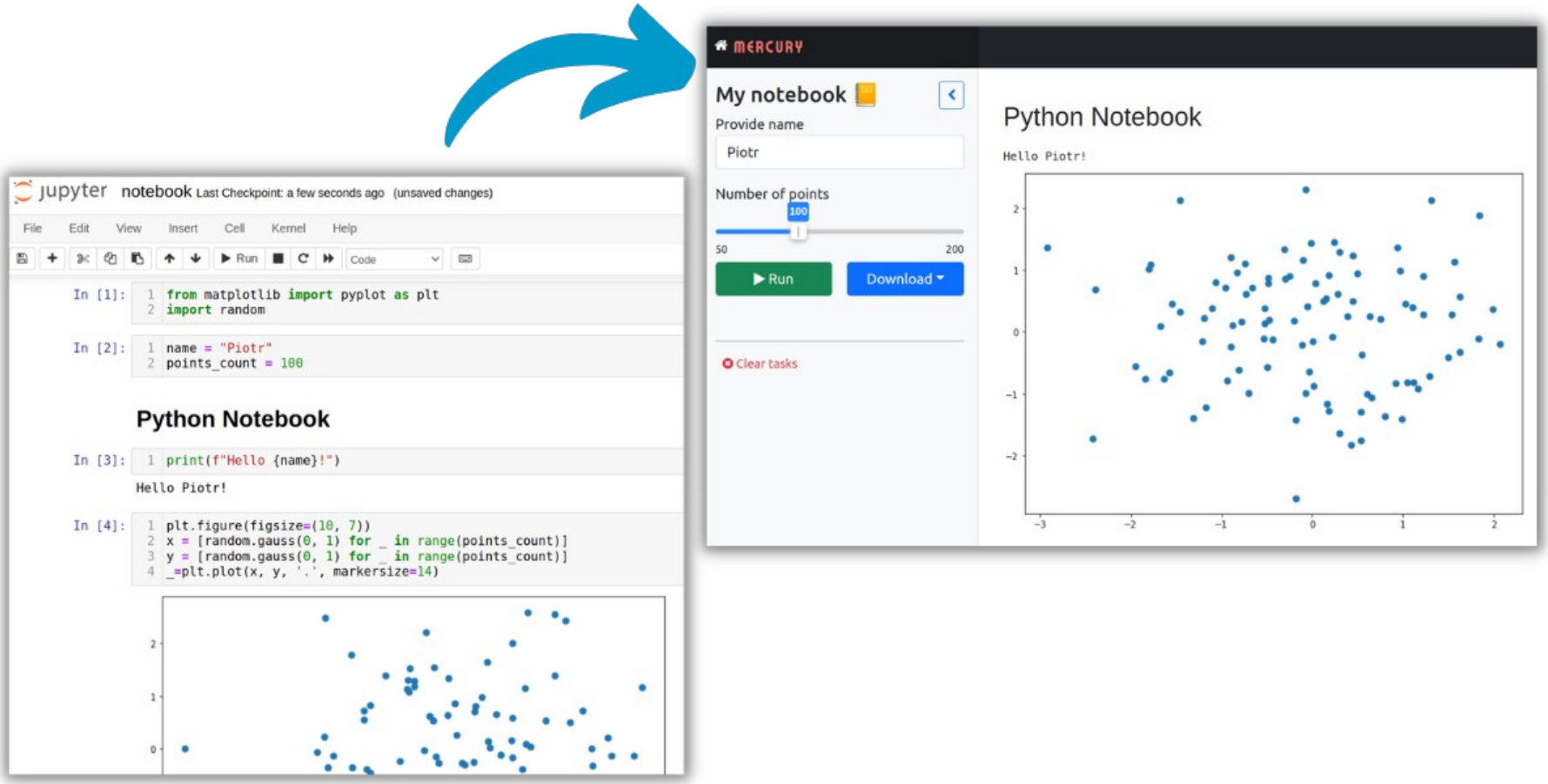
Julia notebooks



Shiny


Shiny from RStudio
Get Started
Gallery
Articles
Reference
Deploy
Help
Contribute
Health expenditure vs. life expectancy, 2008
NEAR NORTH
Route
Superzip
Interactive map
Data explorer
ZIP explorer
Color
Is SuperZIP?
Size
Population
Is SuperZIP?
Centile score
College education
Median income
Population
Interact. Analyze. Communicate.
Take a fresh, interactive approach to telling your data story with Shiny. Let users interact with your data and your analysis. And do it all with R.

Mercury

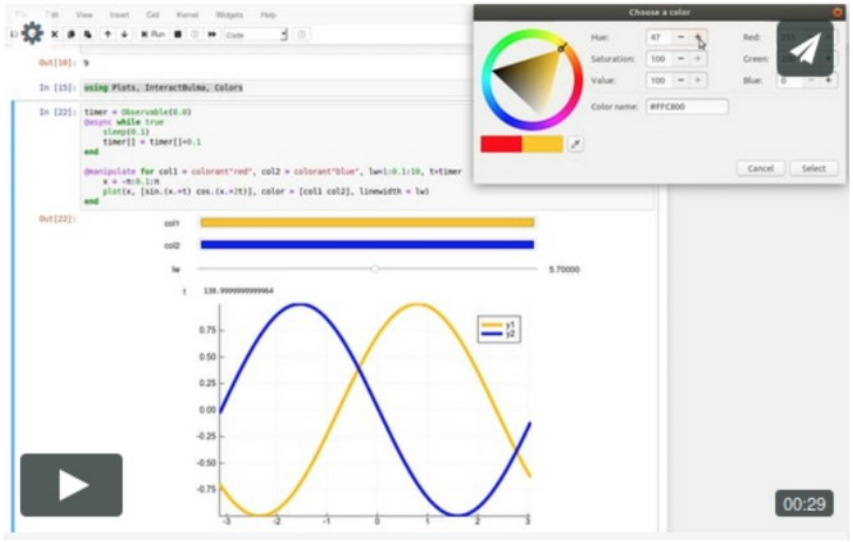

jupyter notebook Last Checkpoint: a few seconds ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Run Code
In [1]:
from matplotlib import pyplot as plt
import random
In [2]:
name = "Piotr"
points_count = 100
Python Notebook
In [3]:
print(f"Hello {name}!")
Hello Piotr!
In [4]:
plt.figure(figsize=(10, 7))
x = [random.gauss(0, 1) for _ in range(points_count)]
y = [random.gauss(0, 1) for _ in range(points_count)]
_=plt.plot(x, y, '.', markersize=14)
MERCURY
My notebook
Provide name
Piotr
Number of points
100
50
200
Run
Download
Clear tasks
Python Notebook
Hello Piotr!

Interact.jl

Interact

build passing docs latest

Interact.jl allows you to use interactive widgets such as sliders, dropdowns and checkboxes to play with your Julia code:

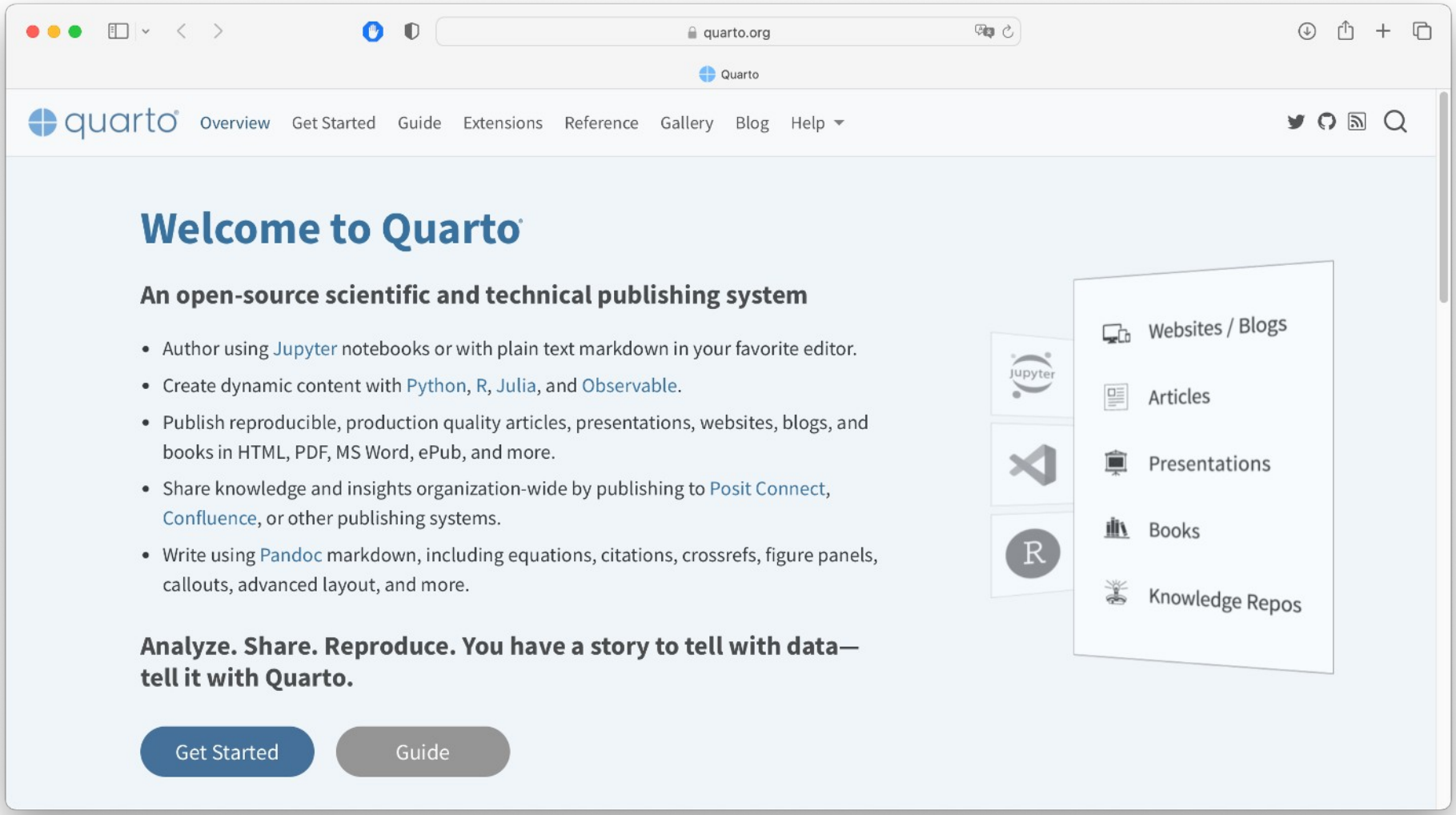


Getting Started

To install Interact, run the following command in the Julia REPL:

```
Pkg.add("Interact")
```

Quarto


quarto.org
Quarto
quarto
Overview
Get Started
Guide
Extensions
Reference
Gallery
Blog
Help
Welcome to Quarto
An open-source scientific and technical publishing system
Author using Jupyter notebooks or with plain text markdown in your favorite editor.
Create dynamic content with Python, R, Julia, and Observable.
Publish reproducible, production quality articles, presentations, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.
Share knowledge and insights organization-wide by publishing to Posit Connect, Confluence, or other publishing systems.
Write using Pandoc markdown, including equations, citations, crossrefs, figure panels, callouts, advanced layout, and more.
Analyze. Share. Reproduce. You have a story to tell with data—tell it with Quarto.
Get Started
Guide
Jupyter
R
Websites / Blogs
Articles
Presentations
Books
Knowledge Repos

4.

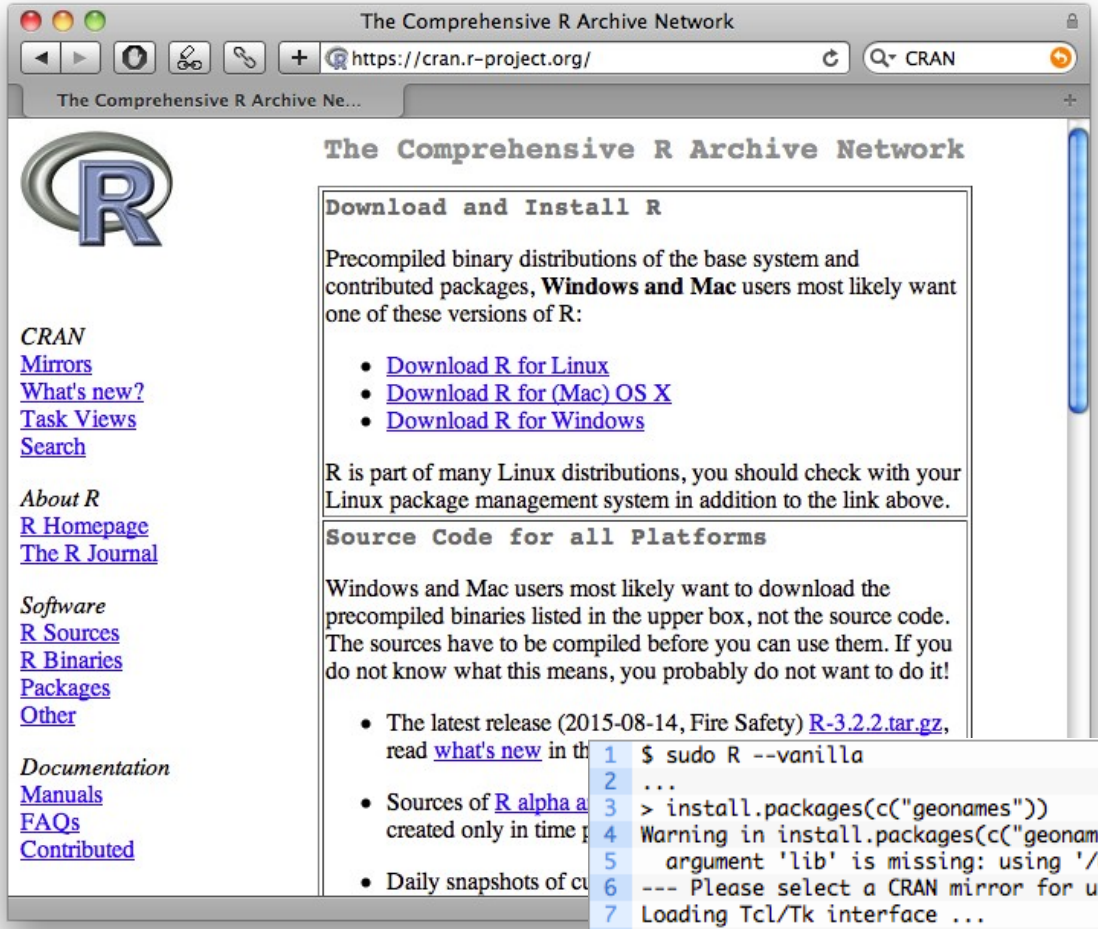
Extensions

Packages – Libraries – Modules

Octave Forge

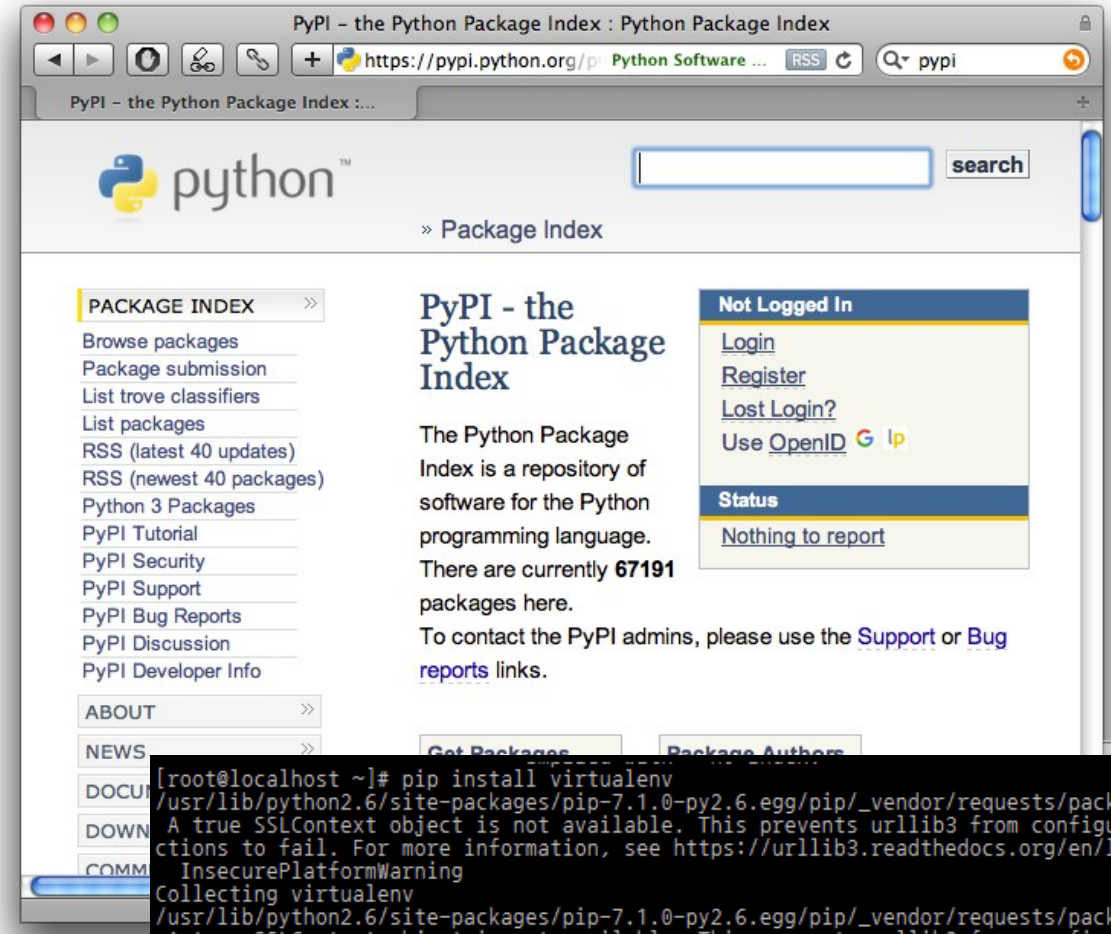

```
>> pkg install -forge image
warning: creating installation directory C:\Octave\Octave-4.0.0\share\octave
warning: called from
 install at line 30 column 5
 pkg at line 405 column 9
For information about changes from previous versions of the image package, r
>> pkg list
Package Name   | Version | Installation directory
---------------+---------+-----------------------
        image |   2.4.0 | C:\Octave\Octave-4.0.0\share\octave\packages\image
```

CRAN



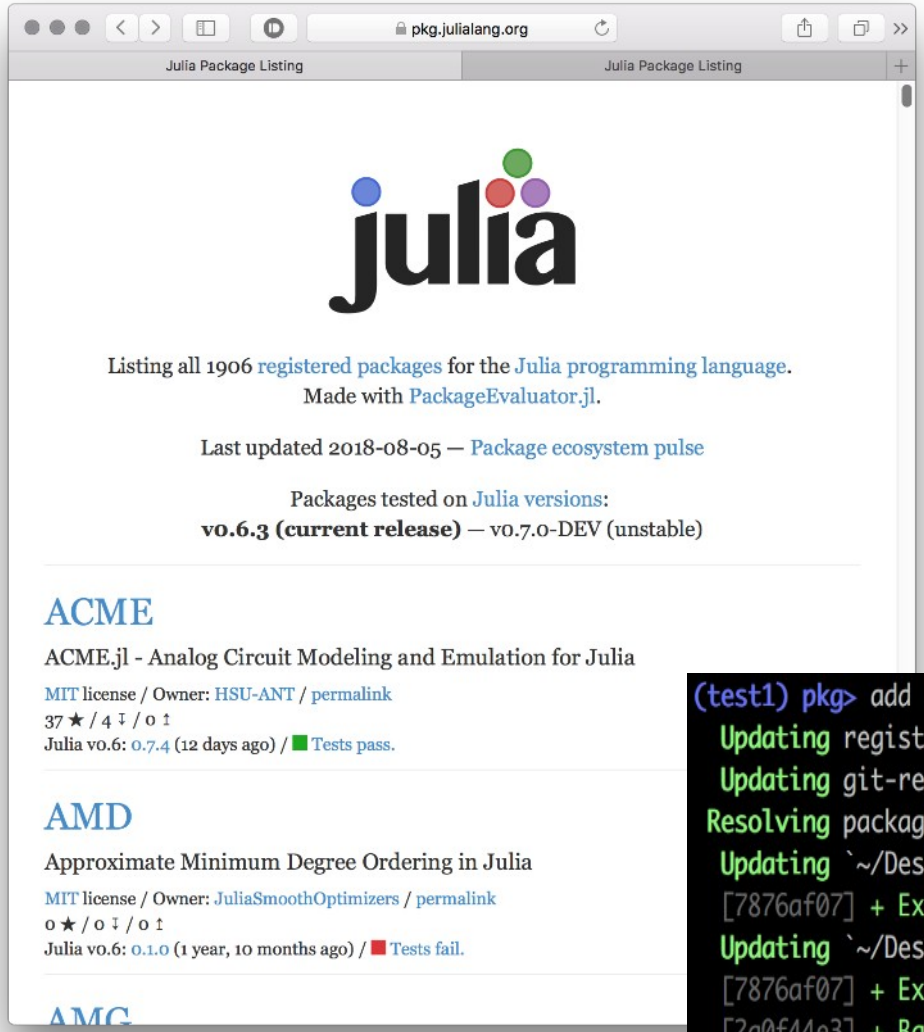

```
$ sudo R --vanilla
...
> install.packages(c("geonames"))
Warning in install.packages(c("geonames")) :
  argument 'lib' is missing: using '/usr/local/lib/R/site-library'
--- Please select a CRAN mirror for use in this session ---
Loading Tcl/Tk interface ...
...
* DONE (geonames)

The downloaded packages are in
        /tmp/Rtmp3FziH3/downloaded_packages
```

PyPI




```
[root@localhost ~]# pip install virtualenv
/usr/lib/python2.6/site-packages/pip-7.1.0-py2.6.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:90: InsecurePlatformWarning:
 A true SSLContext object is not available. This prevents urllib3 from configuring SSL appropriately and may cause certain SSL conne
ctions to fail. For more information, see https://urllib3.readthedocs.org/en/latest/security.html#insecureplatformwarning.
  InsecurePlatformWarning
Collecting virtualenv
/usr/lib/python2.6/site-packages/pip-7.1.0-py2.6.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:90: InsecurePlatformWarning:
 A true SSLContext object is not available. This prevents urllib3 from configuring SSL appropriately and may cause certain SSL conne
ctions to fail. For more information, see https://urllib3.readthedocs.org/en/latest/security.html#insecureplatformwarning.
  InsecurePlatformWarning
  Downloading virtualenv-13.1.0-py2.py3-none-any.whl (1.7MB)
    100% |████████████████████████████████| 1.7MB 201kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-13.1.0
[root@localhost ~]#
```

Julia package ecosystem



Pkg comes with a REPL. Enter the Pkg REPL by pressing `]` from the Julia REPL. To get back to the Julia REPL, press backspace or ^C.

```
(test1) pkg> add Example
  Updating registry at `~/.julia/registries/General`
  Updating git-repo `https://github.com/JuliaRegistries/General.git`
 Resolving package versions...
  Updating `~/Desktop/hobby/julia/test/test1/Project.toml`
  [7876af07] + Example v0.5.1
  Updating `~/Desktop/hobby/julia/test/test1/Manifest.toml`
  [7876af07] + Example v0.5.1
  [2a0f44e3] + Base64
  [8ba89e20] + Distributed
```

5. General tips when it is slow

- Program thoughtfully:
  - Use vectorized functions
  - Avoid loops
  - Preallocate
  - Force type
  - Avoid copy-on-write
- Link to fast libraries (C/C++, Fortran, Java)
- Write low-level parts in C or Fortran
- Compile – jit
- Go parallel

6. Bridges (use ... in ...)

Python → R	http://rpython.r-forge.r-project.org/
Octave → Python	https://pypi.python.org/pypi/oct2py
R → Python	http://rpy.sourceforge.net/
Octave → R	https://cran.r-project.org/web/packages/RcppOctave
Python → Octave	https://github.com/daniel-e/pyoctave
R → Octave	http://www.omegahat.org/ROctave/
R → Julia	https://github.com/Non-Contradiction/JuliaCall
Julia → R	https://github.com/JuliaInterop/RCall.jl
Python → Julia	https://github.com/JuliaPy/pyjulia
Julia → Python	https://github.com/JuliaPy/PyCall.jl

So..

Fast to learn
Fast to code

Challenge

- Look at the files in `/CECI/proj/training/scripting/exercice2` on any CÉCI cluster

```
[dfr@lemaitre3 exercice2]$ pwd
/CECI/proj/training/scripting/exercice2
[dfr@lemaitre3 exercice2]$ ls
res-10.txt  res-18.txt  res-25.txt  res-32.txt  res-3.txt   res-47.txt  res-54.txt  res-61.txt  res-69.txt  res-76.txt  res-83.txt  res-90.txt  res-98.txt
res-11.txt  res-19.txt  res-26.txt  res-33.txt  res-40.txt  res-48.txt  res-55.txt  res-62.txt  res-6.txt   res-77.txt  res-84.txt  res-91.txt  res-99.txt
res-12.txt  res-1.txt   res-27.txt  res-34.txt  res-41.txt  res-49.txt  res-56.txt  res-63.txt  res-70.txt  res-78.txt  res-85.txt  res-92.txt  res-9.txt
res-13.txt  res-20.txt  res-28.txt  res-35.txt  res-42.txt  res-4.txt   res-57.txt  res-64.txt  res-71.txt  res-79.txt  res-86.txt  res-93.txt
res-14.txt  res-21.txt  res-29.txt  res-36.txt  res-43.txt  res-50.txt  res-58.txt  res-65.txt  res-72.txt  res-7.txt   res-87.txt  res-94.txt
res-15.txt  res-22.txt  res-2.txt   res-37.txt  res-44.txt  res-51.txt  res-59.txt  res-66.txt  res-73.txt  res-80.txt  res-88.txt  res-95.txt
res-16.txt  res-23.txt  res-30.txt  res-38.txt  res-45.txt  res-52.txt  res-5.txt   res-67.txt  res-74.txt  res-81.txt  res-89.txt  res-96.txt
res-17.txt  res-24.txt  res-31.txt  res-39.txt  res-46.txt  res-53.txt  res-60.txt  res-68.txt  res-75.txt  res-82.txt  res-8.txt   res-97.txt
[dfr@lemaitre3 exercice2]$ cat res-1.txt
# Result file for experiment
[main]

parameter=0.01
result=0.15492

[meta]
time=531244[dfr@lemaitre3 exercice2]$
```

- We will pretend they are the result of running 100 jobs that take an input parameter and produce output a result (`.INI` file)
- Copy that directory into your home dir

Challenge

- Find for which value of `'parameter'` is `'result'` the lowest.
- Course of action:
  - Read all files and parse them (you might need to install additional packages/libraries/modules)
  - Build two arrays one of parameter values and the other one for result values
  - Remove problematic values (plotting might help here)
  - Find minimum

Challenge

- Pseudo code:
  - “Activate” extension for `.ini` files if necessary
  - Initialize two arrays to hold the values
  - For-loop 1-99 :
    - Read file (using a ready-made extension)
    - Store values in corresponding arrays
  - Remove from array values that show too large a difference between consecutive values (slicing)
  - Find index of minimum value in one array and the corresponding value in the other array

Possible solution

```
# Result file for experiment
[main]

parameter=0.1
result=0.13637

[meta]
time=1407642
```

INI file format

Libraries/packages/modules exist:

- https://nl.mathworks.com/matlabcentral/fileexchange/17177-ini2struct
- https://cran.r-project.org/web/packages/ini/index.html
- https://docs.python.org/3/library/configparser.html
- https://juliapackages.com/p/inifile

Possible solution

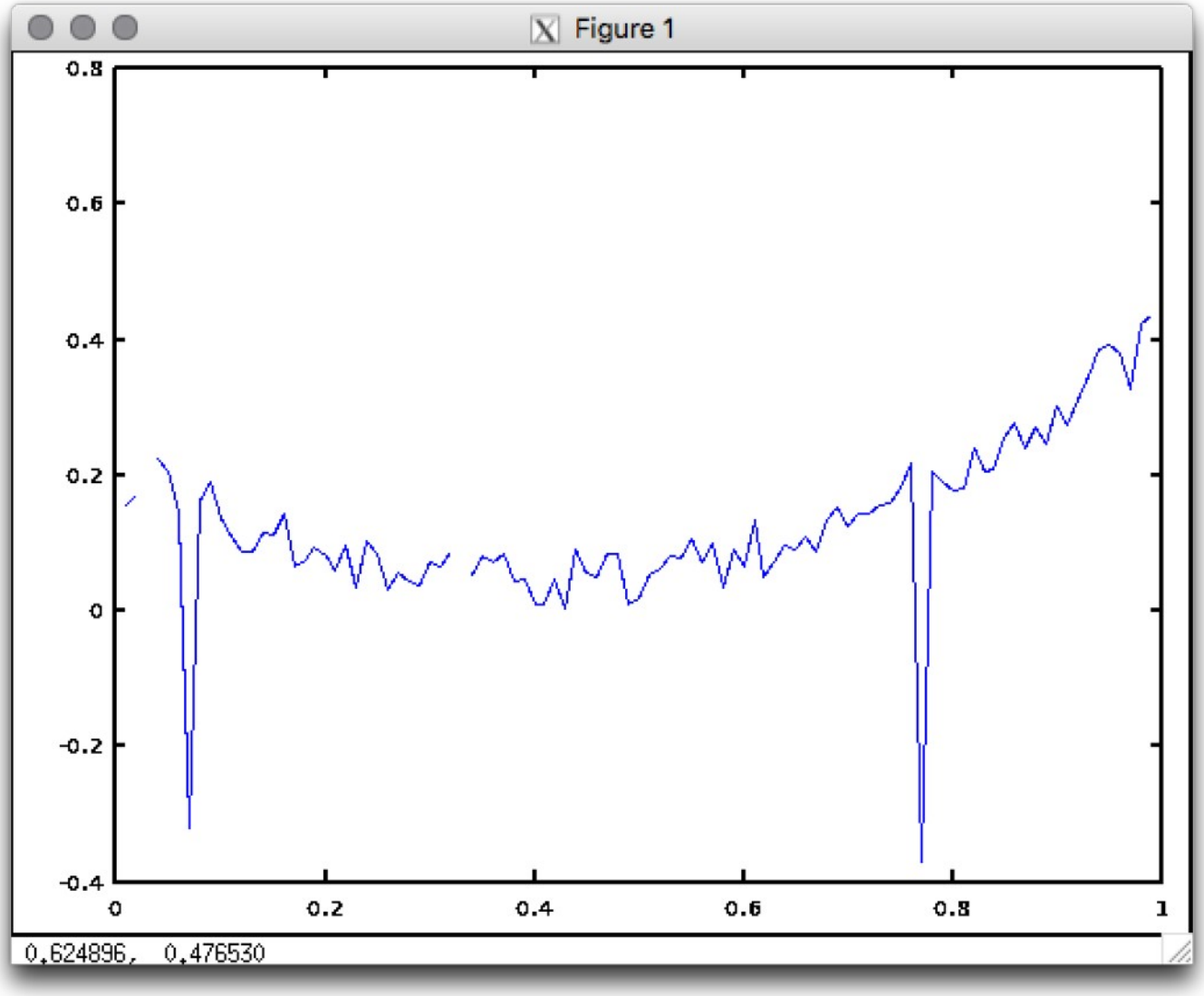
```
nb_res=99;

p=zeros(nb_res,1);
r=zeros(nb_res,1);

for i = 1:nb_res;
  res = ini2struct(sprintf("res-%d.txt", i));
  p(i)=str2double(res.main.parameter);
  r(i)=str2double(res.main.result);
end
r(diff(r)>0.1)=nan;
plot(p,r)
[i, j]=min(r);
i, p(j)
```

```
library(ini)

nb_res <-99

p <- numeric(nb_res)
r <- numeric(nb_res)

for (i in 1:nb_res) {
  f <- read.ini(sprintf('res-%d.txt', i))
  p[i] <- as.numeric(f$main$parameter )
  r[i] <- as.numeric(f$main$result )
}

plot(p,r, 'l')
r[diff(r) > 0.1] <- NA
print(min(r, na.rm=T))
print(p[which.min(r)])
```

```
import configparser
import numpy as np
import matplotlib.pyplot as plt

nb_res = 99

p = np.zeros(nb_res)
r = np.zeros(nb_res)

for i in range(nb_res):
    f = configparser.RawConfigParser()
    f.read("res-{i}.txt".format(i=i+1))
    p[i] = float(f.get('main', 'parameter'))
    r[i] = float(f.get('main', 'result'))


plt.plot(p, r, '-')
r[np.where(np.diff(r) > .1)] = np.nan
print(np.nanmin(r))
print(p[np.nanargmin(r)])
```

```
using IniFile
using Plots

nb_res = 99;

p = Array{Float64}(undef, nb_res)
r = Array{Float64}(undef, nb_res)

for i in 1:nb_res
    ini = read(Inifile(), "res-$i.txt");
    p[i] = parse(Float64, get(ini, "main","parameter"))
    r[i] = parse(Float64, get(ini, "main","result"))
end

r[findall(abs.(r[1:end-1] - r[2:end]).>.1)] .= NaN
r[findall(isnan.(r))] .= Inf
#plot(r)
show(findmin(r))
```

Challenge



Summary

Octave, R, Python* and Julia*

Much programmer-friendly than C/C++/Fortran

Still able to use fast compiled code

Focus on the unsolved problems

Try all and choose one

* upcoming dedicated session!