Building a scientific workbench in Pharo

Konrad Hinsen [1, 2] Serge Stinckwich [3, 4, 5]

[1]Centre de Biophysique Moléculaire, Orléans, France

[2]Synchrotron SOLEIL, Saint Aubin, France

[3]Sorbonne Université, IRD, Unité de Modélisation Mathématiques et Informatique des Systèmes Complexes, UMMISCO, Bondy, France

[4]Université de Yaoundé I, Yaoundé, Cameroon

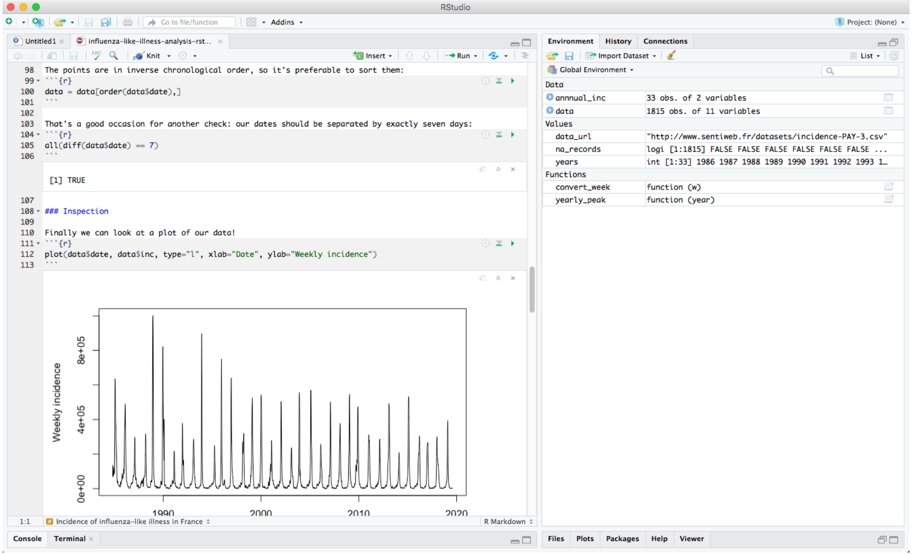
[5]Université de Caen Normandie, Caen, France

27 August 2019

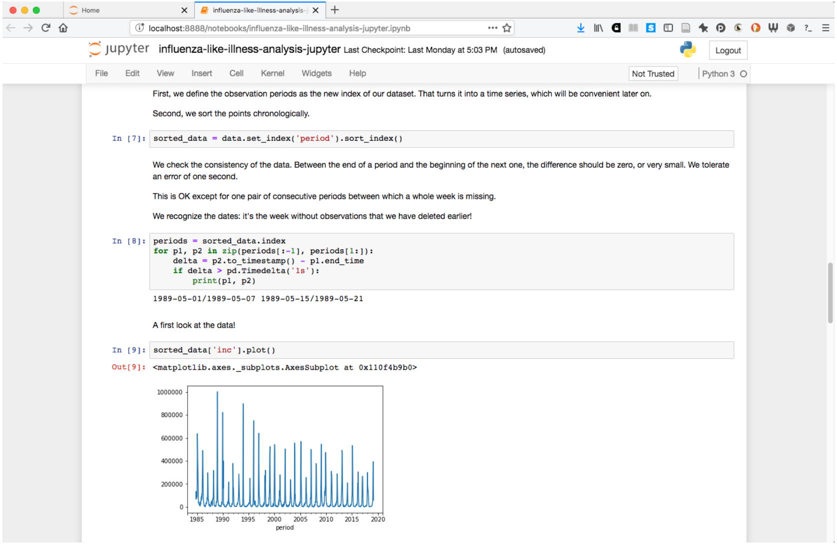
What’s a scientific workbench?

- The IDE of the computational scientist
- Supports the tasks of doing science on a computer:
  - Write and test code
  - Import and export data
  - Process data
  - Perform simulations
  - Inspect experimental and computed data
  - Document all of the above
- Makes computations reproducible.

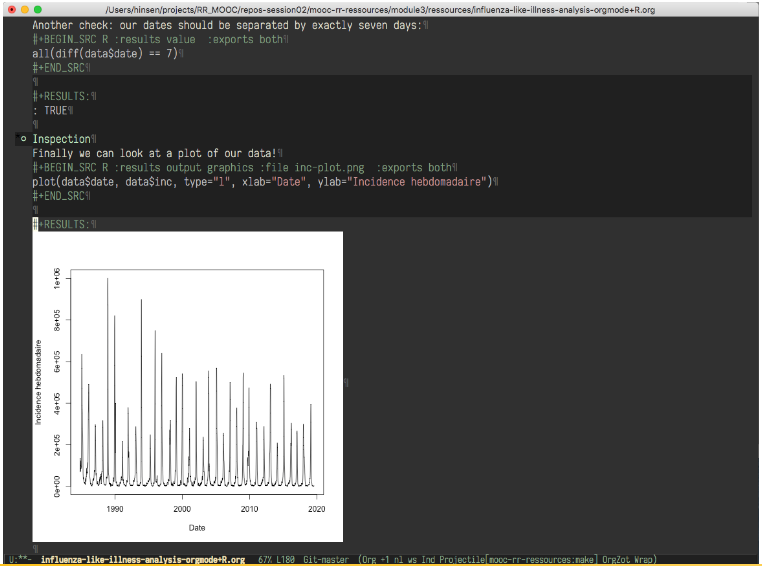
RStudio


RStudio
Go to file/function
Addins
Project: (None)
Untitled1
influenza-like-illness-analysis-rst...
Knit
Insert
Run
98 The points are in inverse chronological order, so it's preferable to sort them:
99 ```{r}
100 data = data[order(data$date),]
101 ```
102
103 That's a good occasion for another check: our dates should be separated by exactly seven days:
104 ```{r}
105 all(diff(data$date) == 7)
106 ```
[1] TRUE
107
108 ### Inspection
109
110 Finally we can look at a plot of our data!
111 ```{r}
112 plot(data$date, data$inc, type="l", xlab="Date", ylab="Weekly incidence")
113 ```
Weekly incidence
0e+00
4e+05
8e+05
1990
2000
2010
2020
1:1
Incidence of influenza-like illness in France
R Markdown
Console
Terminal
Environment
History
Connections
Import Dataset
List
Global Environment
Data
annual_inc
33 obs. of 2 variables
data
1815 obs. of 11 variables
Values
data_url
"http://www.sentiweb.fr/datasets/incidence-PAY-3.csv"
na_records
logi [1:1815] FALSE FALSE FALSE FALSE FALSE FALSE ...
years
int [1:33] 1986 1987 1988 1989 1990 1991 1992 1993 1…
Functions
convert_week
function (w)
yearly_peak
function (year)
Files
Plots
Packages
Help
Viewer

Jupyter


Home
influenza-like-illness-analysis-
localhost:8888/notebooks/influenza-like-illness-analysis-jupyter.ipynb
jupyter influenza-like-illness-analysis-jupyter Last Checkpoint: Last Monday at 5:03 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
First, we define the observation periods as the new index of our dataset. That turns it into a time series, which will be convenient later on.
Second, we sort the points chronologically.
In [7]: sorted_data = data.set_index('period').sort_index()
We check the consistency of the data. Between the end of a period and the beginning of the next one, the difference should be zero, or very small. We tolerate an error of one second.
This is OK except for one pair of consecutive periods between which a whole week is missing.
We recognize the dates: it's the week without observations that we have deleted earlier!
In [8]: periods = sorted_data.index
for p1, p2 in zip(periods[:-1], periods[1:]):
delta = p2.to_timestamp() - p1.end_time
if delta > pd.Timedelta('1s'):
print(p1, p2)
1989-05-01/1989-05-07 1989-05-15/1989-05-21
A first look at the data!
In [9]: sorted_data['inc'].plot()
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x110f4b9b0>
1000000
800000
600000
400000
200000
0
1985 1990 1995 2000 2005 2010 2015 2020
period

Emacs


/Users/hinsen/projects/RR_MOOC/repos-session02/mooc-rr-ressources/module3/ressources/influenza-like-illness-analysis-orgmode+R.org
Another check: our dates should be separated by exactly seven days:
#+BEGIN_SRC R :results value :exports both
all(diff(data$date) == 7)
#+END_SRC
#+RESULTS:
: TRUE
Inspection
Finally we can look at a plot of our data!
#+BEGIN_SRC R :results output graphics :file inc-plot.png :exports both
plot(data$date, data$inc, type="l", xlab="Date", ylab="Incidence hebdomadaire")
#+END_SRC
#+RESULTS:
Incidence hebdomadaire
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
1990
2000
2010
2020
Date
U:**- influenza-like-illness-analysis-orgmode+R.org 67% L180 Git-master (Org +1 nl ws Ind Projectile[mooc-rr-ressources:make] OrgZot Wrap)

Shifting priorities

Traditional focus: *get work done efficiently*

- interactive computation
- generate plots and tables (for pasting into publications produced outside of the workbench)

More recent criteria: *robust and understandable results*

- reproducible computations
- shared/publishable raw datasets
- well-documented computations
- document while you compute

The state of the art: computational notebooks

A fusion of scripts, REPLs, and literate programming, invented in the 1980's by Mathematica

- A linear sequence of so-called "cells"
- Three cell types:
  - Text cells hold rich text for documentation
  - Code cells contain code snippets
  - Output cells show the output of one code snippet (text or graphics)
- Code cells can be executed one by one, manually...
- ... or sequentially as part of a whole-notebook execution.

Many implementations: Mathematica, Jupyter, R Markdown, Emacs/Org-Mode, ...

Limitations of notebooks

- Linear sequence of cells: no way to structure or modularize
- Made worse by shared mutable state...
- ... and even worse by interactive cell execution.
- Documentation follows code structure: no way to relegate technical details to an appendix
- Data dependencies are not explicit, nor easily visible.
- Neither code nor data are reusable by other notebooks.
- Different tools/user interfaces for notebooks and library code.

Notebooks blissfully ignore decades of software engineering achievements.

Smalltalk to the rescue

Hypothesis:

> A Smalltalk system is a much better starting point
> for designing a scientific workbench than a REPL.

Nice properties:

- well-known to this audience!

Missing pieces:

- A documentation tool that allows embedding code and data.
- Management of computational tasks and data dependencies to replace the notebook's linear control flow
- Support libraries for scientific computing.

Nothing new

Mathematics and Computers in Simulation 31 (1989) 371–381 371
North-Holland

SMALLTALK – THE NEXT GENERATION SCIENTIFIC COMPUTING INTERFACE?

Richard L. PESKIN, Sandra S. WALTHER and Andy M. FRONCIONI

*Center for Computer Aids For Industrial Productivity, Rutgers University *, P.O. Box 1390, Piscataway, New Jersey 08855-1390, U.S.A.*

The need for rapid prototyping of numerical simulations is considered, and an object-oriented, graphical based system (Smalltalk) is proposed as a basis for a new approach to user interfaces for scientific computing. The interface system requirements for problem expression, automatic programming, visualization, computational steering, and concurrent computing are discussed.

1. Introduction

While scientific and engineering computation needs have been a major driving force in the

Glamorous Toolkit

Nice properties:

- presented yesterday to this audience
- specifically for a scientific workbench: an excellent documentation tool

Missing pieces:

- Management of computational tasks and data dependencies
  to replace the notebook's linear control flow

ActivePapers

A research project about performing and communicating computer-aided research

- Started in 2011.
- Initial focus: reproducible high-performance computing.
- Management of computational tasks and data dependencies
- Current implementation based on Python...
- ... and a lousy user interface: very basic CLI

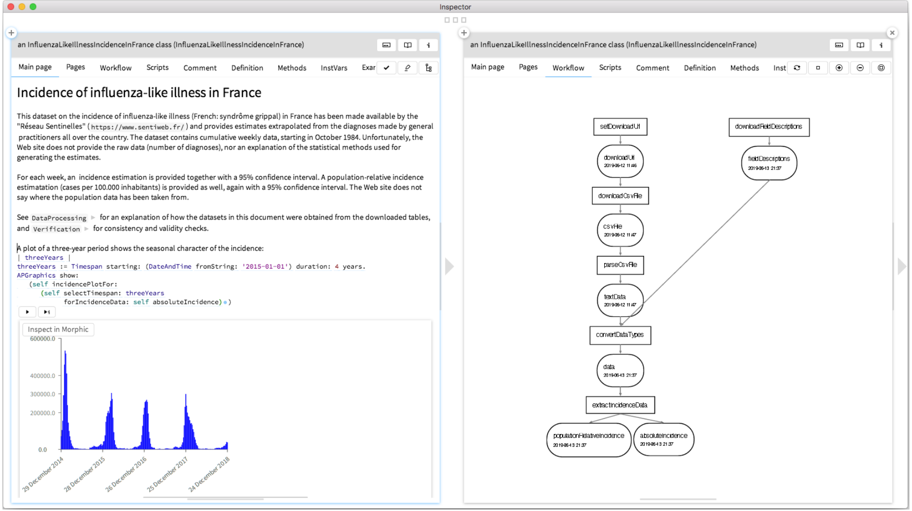
PolyMath

- Release 1.0 last week
- 300 classes, 50 packages, 24K LOC, 806 unit tests
- Ordinary differential Equations, Random Number Generators, Linear algebra, Matrices, Complex Numbers, FFT, Polynomials, Probability distributions, ...
- more recently: Automatic differentiation, Principal Component Analysis, t-SNE,
- DataFrame to do data analysis
- Talk on PolyMath next thursday

Working hypothesis

Pharo + Glamorous Toolkit + PolyMath + ActivePapers
= Scientific Workbench

Demo


Inspector
an InfluenzaLikeIllnessIncidenceInFrance class (InfluenzaLikeIllnessIncidenceInFrance)
Main page
Pages
Workflow
Scripts
Comment
Definition
Methods
InstVars
Incidence of influenza-like illness in France
This dataset on the incidence of influenza-like illness (French: syndrôme grippal) in France has been made available by the "Réseau Sentinelles" (https://www.sentiweb.fr/) and provides estimates extrapolated from the diagnoses made by general practitioners all over the country. The dataset contains cumulative weekly data, starting in October 1984. Unfortunately, the Web site does not provide the raw data (number of diagnoses), nor an explanation of the statistical methods used for generating the estimates.
For each week, an incidence estimation is provided together with a 95% confidence interval. A population-relative incidence estimatation (cases per 100.000 inhabitants) is provided as well, again with a 95% confidence interval. The Web site does not say where the population data has been taken from.
See DataProcessing for an explanation of how the datasets in this document were obtained from the downloaded tables, and Verification for consistency and validity checks.
A plot of a three-year period shows the seasonal character of the incidence:
| threeYears |
threeYears := Timespan starting: (DateAndTime fromString: '2015-01-01') duration: 4 years.
APGraphics show:
(self incidencePlotFor:
(self selectTimespan: threeYears
forIncidenceData: self absoluteIncidence))
Inspect in Morphic
600000.0
400000.0
300000.0
200000.0
0.0
29 December 2014
28 December 2015
26 December 2016
25 December 2017
24 December 2018
setDownloadUrl
downloadUrl
downloadCsvFile
csvFile
parseCsvFile
textData
convertDataTypes
data
extractIncidenceData
populationRelativeIncidence
absoluteIncidence
downloadFieldDescriptions
fieldDescriptions

Outlook

A lot of work remains to be done:

- More domain-specific libraries
- Interfaces to other languages
- Data management outside of the Pharo image
- Publishing ActivePapers on the Web