LIS

CROSS-NATIONAL
DATA CENTER
in Luxembourg

Julia as a software for Official Statistics and Social Sciences

Josep Espasa Reig

Data Scientist

This presentation

- ❑ What is Julia and why could it be useful for Official Statistics and Social Sciences?
- ❑ A few thoughts on adding another software to an organization’s toolkit
- ❑ Benchmarks of Julia vs R code
- ❑ Assessment of Julia package ecosystem maturity

What is the problem?

- ❑ R and Python are slow languages
- ❑ Typically complemented with low-level languages (e.g. C, C++ or Rust)
- ❑ These are difficult languages to learn and code with!

What is Julia?

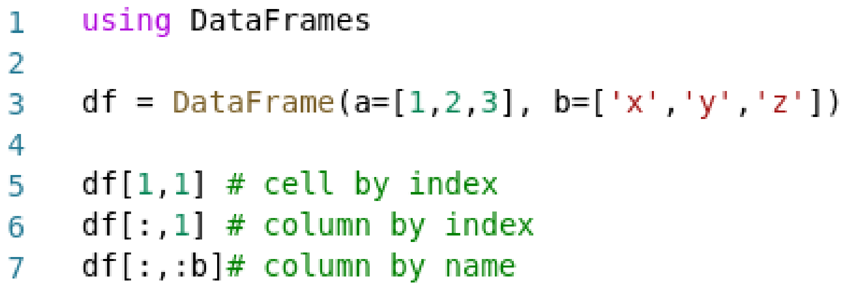
- An open-source, dynamically typed language (like R and Python)
- Uses Just in time (JIT) Compilation
- Syntactically similar languages
- Julia feels more modern, easier to read and cleaner (personal opinion)
- R and Python have packages to run Julia code (and vice versa)
- Cons: the package ecosystem does not have the same maturity than the R and Python ones (see assessment slides)

Cost-benefit analysis

- ❑ Adding another software to a DS team has costs:
  - ❑ *Skills*
  - ❑ *Development*
  - ❑ *Maintenance*
- ❑ It should also have benefits:
  - ❑ *Speed (see benchmarks slide in a minute)*
  - ❑ *Better features (e.g. multiple dispatch)*

Similarity between Julia, R and Python

Julia

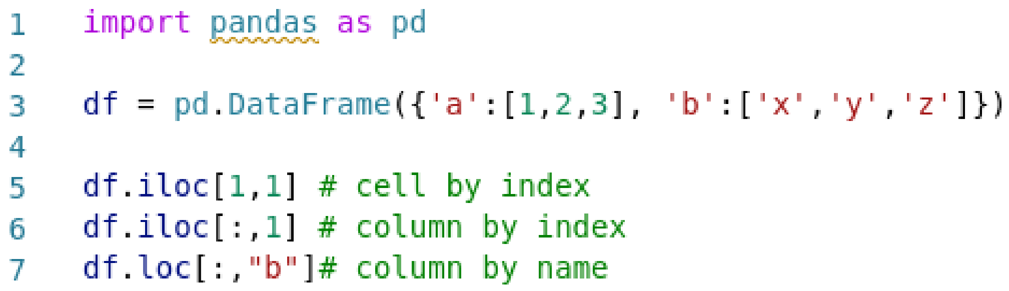


```julia
using DataFrames

df = DataFrame(a=[1,2,3], b=['x','y','z'])

df[1,1] # cell by index
df[:,1] # column by index
df[:,:b]# column by name
```

Python

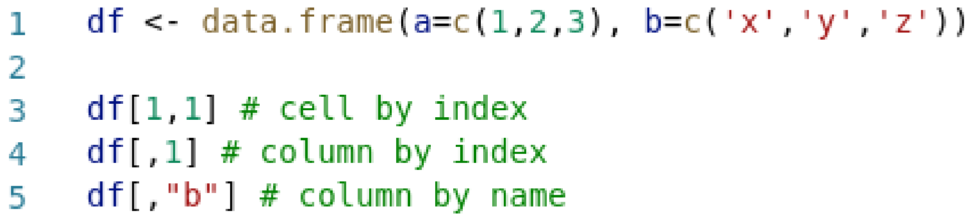


```python
import pandas as pd

df = pd.DataFrame({'a':[1,2,3], 'b':['x','y','z']})

df.iloc[1,1] # cell by index
df.iloc[:,1] # column by index
df.loc[:,"b"]# column by name
```

R



```r
df <- data.frame(a=c(1,2,3), b=c('x','y','z'))

df[1,1] # cell by index
df[,1] # column by index
df[,"b"] # column by name
```

Benchmarks

- *Compared Julia and R performance on:*
  - Inequality indicators: Gini, Atkinson, Foster–Greer–Thorbecke (FGT), poverty headcount, poverty gap, Watts poverty index, Theil poverty index, mean log deviation (MLD)
  - Bootstrap estimates of the same indicators (1000 resamples)
  - Bootstrap estimates with a ‘grouped by’ variable (split-apply-combine).
- *Overhead running Julia functions from R*
- *To reproduce the benchmarks: you can find the repositories with Dockerfiles here:*
  - https://github.com/JosepER/ntts_2023_benchmarking_r
  - https://github.com/JosepER/ntts_2023_benchmarking_julia

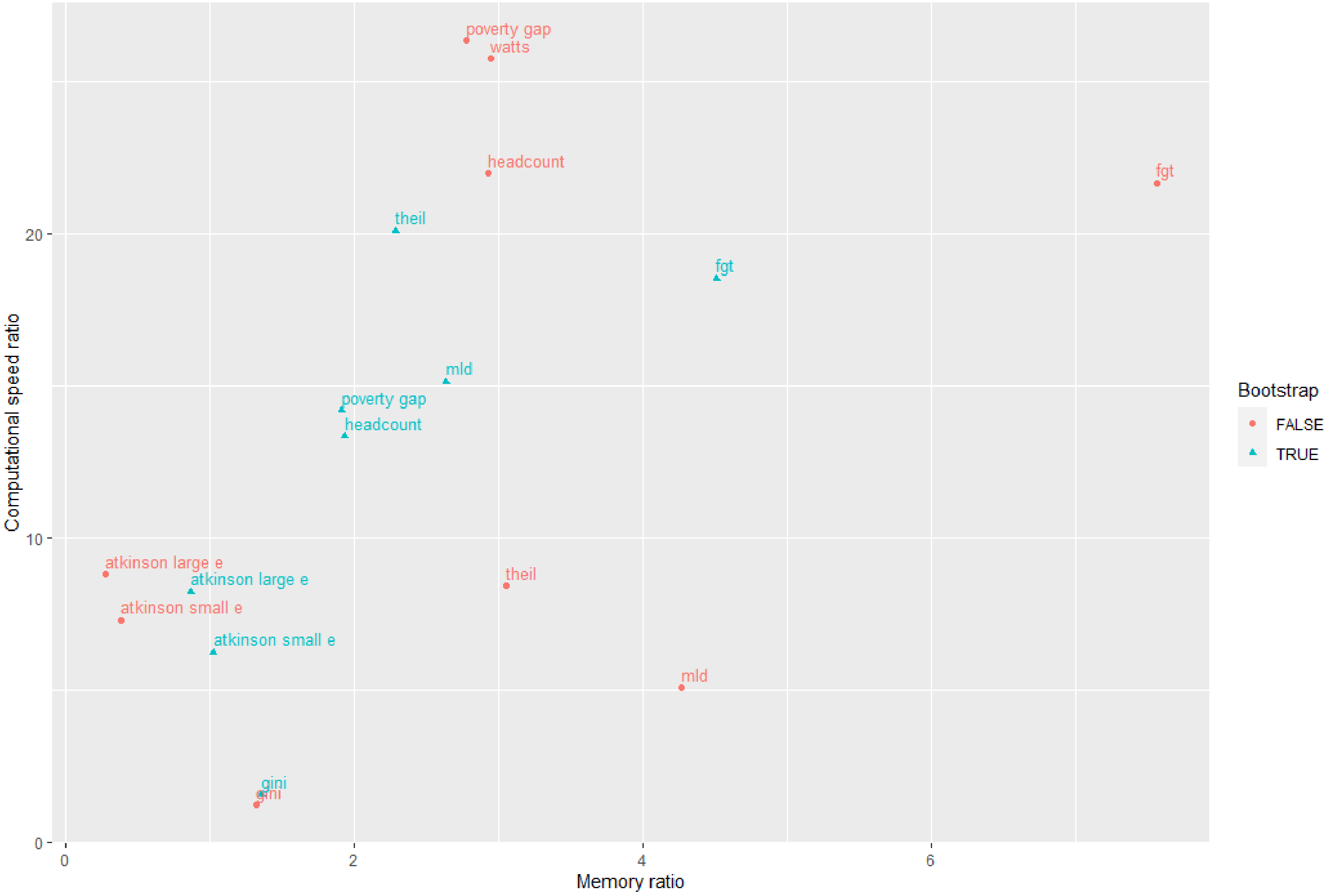
Benchmarks

Function	R (seconds)	Julia (seconds)	Ratio
Gini	0.020	0.017	1.20
Atkinson (ε > 1)	0.022	0.03	8.79
Atkinson (ε < 1)	0.015	0.002	7.29
FGT	0.130	0.006	22.2
Headcount	0.121	0.005	22.0
Poverty Gap	0.147	0.006	26.4
Watts	0.157	0.006	25.8
Theil	0.011	0.001	8.41
MLD	0.011	0.002	5.06

Benchmarks

Function (with bootstrap M=1000)	R (seconds)	Julia (seconds)	Ratio
Gini	29.7	19.3	1.54
Atkinson (ε > 1)	32.96	4.01	8.22
Atkinson (ε < 1)	22.32	3.59	6.22
FGT	154.19	8.32	18.5
Headcount	139.95	10.5	13.3
Poverty Gap	150.41	10.6	14.2
Watts	123	1.66	74.1
Theil	63.7	3.17	20.1
MLD	64.7	4.28	15.1
MLD* (grouped by htype)	137	4.8	28.54

Benchmarks



poverty gap
watts
headcount
fgt
theil
fgt
mld
poverty gap
headcount
atkinson large e
atkinson large e
atkinson small e
theil
atkinson small e
mld
gini
gini
Computational speed ratio
20
10
0
0
2
4
6
Memory ratio
Bootstrap
● FALSE
▲ TRUE

Overhead benchmarks

Function	Julia called from R (seconds)	Julia (seconds)	Ratio
Gini	0.019	0.017	1.2
Atkinson (ε > 1)	0.0085	0.003	3.4
Atkinson (ε < 1)	0.008	0.002	3.8
Theil	0.006	0.0013	4.5
MLD	0.0048	0.0022	2.2

Maturity of Julia packages

- *Could a team of DS use Julia for Official Statistics and Social Sciences tasks?*
- *Analyzed the packages in the following areas:*
    - Importing data from datasets
    - Interacting with SQL databases
    - Manipulation of tabular datasets
    - Sampling and sample survey planning
    - Statistical matching
    - Weighting and calibration of survey samples
    - Imputation and treatment of missing values
    - Variance estimation for complex survey designs
- *Classified into 3 categories:*
    - Mature
    - Partially available/developing
    - Not available

Maturity of Julia packages

Maturity	Area
Mature	Importing data from datasets
Mature	Interacting with SQL databases
Mature	Manipulation of tabular datasets
Partially available/developing	Sampling and sample survey planning
Not available	Statistical matching
Partially available/developing	Weighting and calibration of survey samples
Partially available/developing	Imputation and treatment of missing values
Partially available/developing	Computation of statistical estimates and variance estimation

Conclusions

- ❑ *Using Julia can lead to substantial speed increases in certain processes (typically from 2x to 20x).*
- ❑ *There should also be a reduction in memory use, but more moderate.*
- ❑ *Julia has a relatively mature package ecosystem for general tasks, but lacks tools for more specific ones.*

Thank you!

EspasaReig@lisdatacenter.org

Presentation and full repository at:

github.com/JosepER/ntts2023_julia_for_official_statistics