Towards a new architecture model for cooperation using open and shared algorithms

J.-M. Museux & J. Grazzini

November, 6th – 7th 2017

New Challenges for Statistical Software
Outline: from high-level vision to low-level implementation and issues...

- European context for shared software
- **R@Eurostat**
- Beyond *R* and open software, open algorithms
- Consequences for architecture of open software
- Conclusion
• **European context for shared software:** *ESS* and other acronyms, and few keywords

• *R@Eurostat*

• Beyond *R* and open software, open algorithms

• Consequences for architecture of open software

• Conclusion
European context – ESS

**political**: DSM (Digital Single Market)

**statistical**: European Statistical System (ESS)

- NSIs are committed to **sharing and reuse**
- NSIs and Eurostat participate in **standards building and maintenance** (CSPA, GSBPM...
CSPA as enabler for shared services

Different tools and processes are developed for the same requirements: high cost, high redundancy...
CSPA as enabler for shared-services

Share and reuse of “plug-and-play” services: low cost, low redundancy,…
ESS RAF: scenarios for CSPA

**Shared**
Application services are shared physically between NSIs by having a **single instance of service**.

**Replicated**
Application services are shared physically between NSIs by **deploying solution locally**.

**Coordinated**
Standards for statistical methodology and processes are jointly defined and agreed upon through **interoperability specification**.
ESS SPRA & SERV: specification & implementation of CSPA

- Services modeled to expose **generic application functions**

- **Standardised business description** of services so as to ensure compliance

- Domain-specific business logic defined at the level of an **orchestration component**

- **Service catalogue**

- Replicated or shared services made available via a **central service oriented platform**
outline

- European context for shared software
- **R@Eurostat**: current status and foreseen evolution
- Beyond $R$ and open software, open algorithms
- Consequences for architecture of open software
- Conclusion
SWOT analysis of R adoption @Eurostat

**Strengths**
- reduced costs of development
- increased interoperability/standardisation
- collaborative developments

**Weaknesses**
- still long development cycle
- lack of critical mass of services
- cost of integration/migration

**Opportunities**
- algorithms marketplace
- technological developments
- cloud and DaaS platform solutions

**Threats**
- architecture not aligned (legacy)
- preference to code reuse and open source (dependencies)
- calls for a new type of architecture
Considerations for R development@Eurostat

• There is not necessarily one-to-one matching between statistical processes, IT modules and R packages

• R packages / IT modules: trade-off between
  - **scope of application** (flexibility)
  - **ease of application** (reusability)
  - efficiency
  - simplicity

• Statistical processes: degree to which processes are configured and parameterised?

• Preference for **deployment of simple and small** modules in which the statistical processes (packages?) are encapsulated
Scenarios of R deployment@Eurostat

**Statistical research**
distributed locally for explorative research

**Ad-hoc data analysis**
deployed as a service for ad-hoc studies

**Statistical production**
deployed as a service in production environments

Statistical data hub (data virtualisation)
outline

- European context for shared software
- R@Eurostat

- Beyond R and open software, open algorithms: more keywords and some considerations

- Consequences for architecture of open software
- Conclusion
Open algorithm framework (1/4)

- "Open algorithms" guarantee:
  - Transparency
  - Reproducibility
  - Reusability
  - Verifiability

- "Open software" are obviously preferred – though also susceptible to downside...

- "Open source code" is more suited
Open algorithm framework (2/4)

Comprehensiveness of the statistical information, e.g.:
- what (method/technique) is actually implemented?
- is that done consistently? black-box?

+ yesterday example: mu-argus vs. R sdc (cf. Meindl’s presentation)

Example: quantile, Gini, aggregates...

Chapter 2
More Than a Dozen Alternative Ways of Spelling Gini

What methods are used to calculate aggregates for groups of countries?

J. Hyndman and Yanan Fan

Sample Quantiles in Statistical Packages

There are a large number of different definitions used for sample quantiles in statistical computer packages. Often with other than the midranks definition will be adopted.

The quantile (p) = \((1 - \gamma)X_{(j)} + \gamma X_{(j+1)}\)

where \(\frac{j - m}{n} \leq p < \frac{j - m + 1}{n}\) (1)

Revise methodological choices...
Open algorithm framework (3/4)

Control of the statistical implementation, *e.g.*:
- what is the default configuration?
- how is the operation set up?
- what are the ad-hoc parameters?

example: quantile implementations

Ensure/enforce configuration/parameters settings...
Open algorithm framework (4/4)

**Maintenance** (continuity and/or migration) of operations, e.g.:
- did we keep a track of producer's actions (logs)?
- can we regenerate the outputs?
- is the consistency of the outputs guaranteed (platform dependence)?

*example: sampling*

Deal with technological constraints...
Examples of (recent, in-house) open algorithms @Eurostat

- **PING** (SAS): generation of derived products on income and living conditions

- **EuroGeoStat** (Java): generic handling and management of geographical and statistical data

- **ICW2** (R): creation of experimental statistics on income, consumption and wealth

- **quantile** (SAS/R/Python/C): software-agnostic and harmonised implementation of various quantile algorithms

- ...
outline

- European context for shared software
- R@Eurostat: current status and foreseen evolution
- Beyond R and open software, open algorithms

**Consequences for architecture of open software:** less keywords and more considerations

- Conclusion
New architecture model

- **Agile (and flexible):** integrates *quickly and continuously*

- **Granular:** starts small (still, not from scratch...), builds **complex operations from simple and small parts**

- **Modular:** incorporates *modular and customisable IT components* that encapsulate statistical methods

- **Agnostic:** releases *constraint* on programming language/software
Did we say “agile”? 

- sharing & openness (transparency) 
- verifiability & collaboration (inspection) 
- reusability & reproducibility (adaptation)
outline

- European context for shared software
- R@Eurostat: current status and foreseen evolution
- Beyond R and open software, open algorithms
- Consequences for architecture of open software

- Conclusion: same keywords, new issues
Conclusion: open algorithms...

- **vision:**
  - Users/producers shall become "produsers"

- **model:**
  - **Open**, **shared**, and **collaborative**
  - **Reusable** and **verifiable**
  - **Agile**, **flexible**, and **continuous**

- **practice:**
  - Develop libraries of **validated statistical methods and IT components** (R...but not only)
  - Gain **experience from deployment**
... and open issues

- processes (development):
  - Testing and certification of statistical algorithms (sound methodology) and IT components (efficient implementation)?
  - Quality control and assessment (actors: NSIs, community, CRAN, ...)?
  - Maintenance of releases and versioning (governance)?

- system (deployment):
  - Integration of multiple data source and workflows?
  - Automation and transition (migration) from research-grade experiments to corporate production?
  - Audit trail?