traffic tutorial

a toolbox for processing and analysing air traffic data

Xavier Olive, ONERA

20th June, 2022

Agenda

Virtual session: not a hands-on tutorial this time

1. General introduction
    - History: why a traffic library?
    - Overall structure of the library
    - Going through the documentation
2. Quick live coding session
3. Perspectives

History

Started early 2018, based on two frustrations:

- how to download data from OpenSky Impala shell
- a lot of boilerplate one shot code to copy/paste between two papers

and one question:

- how did you make those maps? (hold my beer...)

Frustration with existing tools:

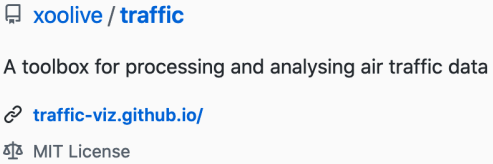
- `pandas` misses semantics for trajectories
- `geopandas` suits well statical geometrical shapes, but not time series

Overview of the library

- Aeronautical data
- Core structures
- Reference datasets
- Visualisation facilities

Essentially, a semantic built around trajectories and set of trajectories.

References



- Code: https://github.com/xoolive/traffic/
  Documentation: https://traffic-viz.github.io/
- Started early 2018
- traffic, a toolbox for processing and analysing air traffic data, *Journal of Open Source Software* (4), 2019. DOI: 10.21105/joss.01518

The future of the library

Community effort, call for contributions!!

- Provide more reference datasets, enrich with metadata
- Standardise a grammar of definitions for aircraft trajectory processing (language agnostic)

Coming soon!

- A Javascript adapter for data visualisation on the web:
  https://observablehq.com/collection/@xoolive/open-aviation
- Pattern detection (point merge, holding patterns) and environmental impact with OpenAP
- More scalable executions with Spark
- More Docker adapters to facilitate the installation

Key take-aways

- Open-source is better than closed source
  … but it does not mean it is perfect ××× DISCLAIMER ×××
- Data has no immediate value
- Information is valuable, but the extraction process is hard
- Code is not precious, expertise is
- Humans read code, help experts read it too
- Use declarative style for better reproducibility