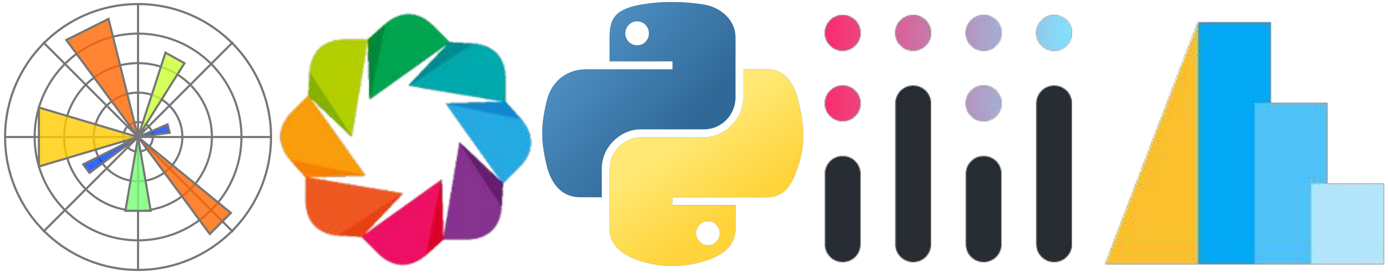
The Python Data Visualisation Landscape



In 2023

fosstodon.org/@owenrlamont

This meeting could’ve been an ~~email~~ website

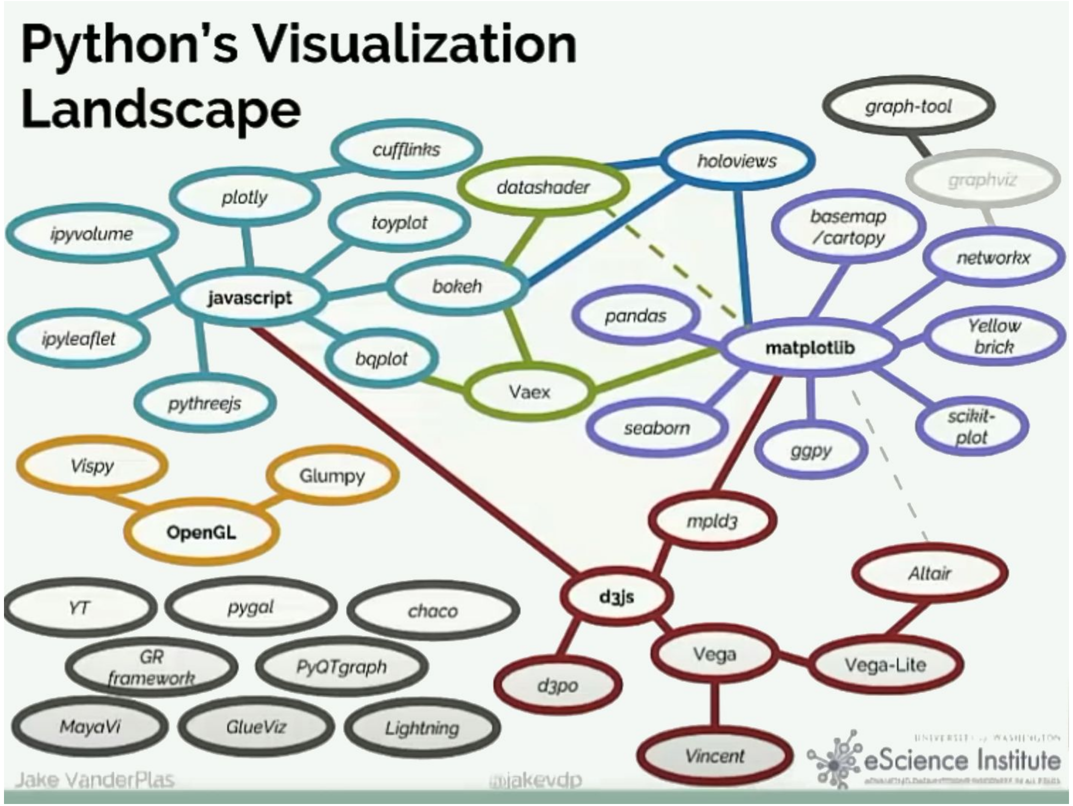
Actually it is, the PyViz site maintains list of data visualisation packages across the Python ecosystem and their usage/health:

https://pyviz.org

This originally focused on HoloViz tools (I believe maintained by Anaconda?) but has been generalised to cover all Python data visualisation tooling and now HoloViz specific packages are documented at:

https://holoviz.org

Jake’s Talk from 2017 - How much is still relevant?



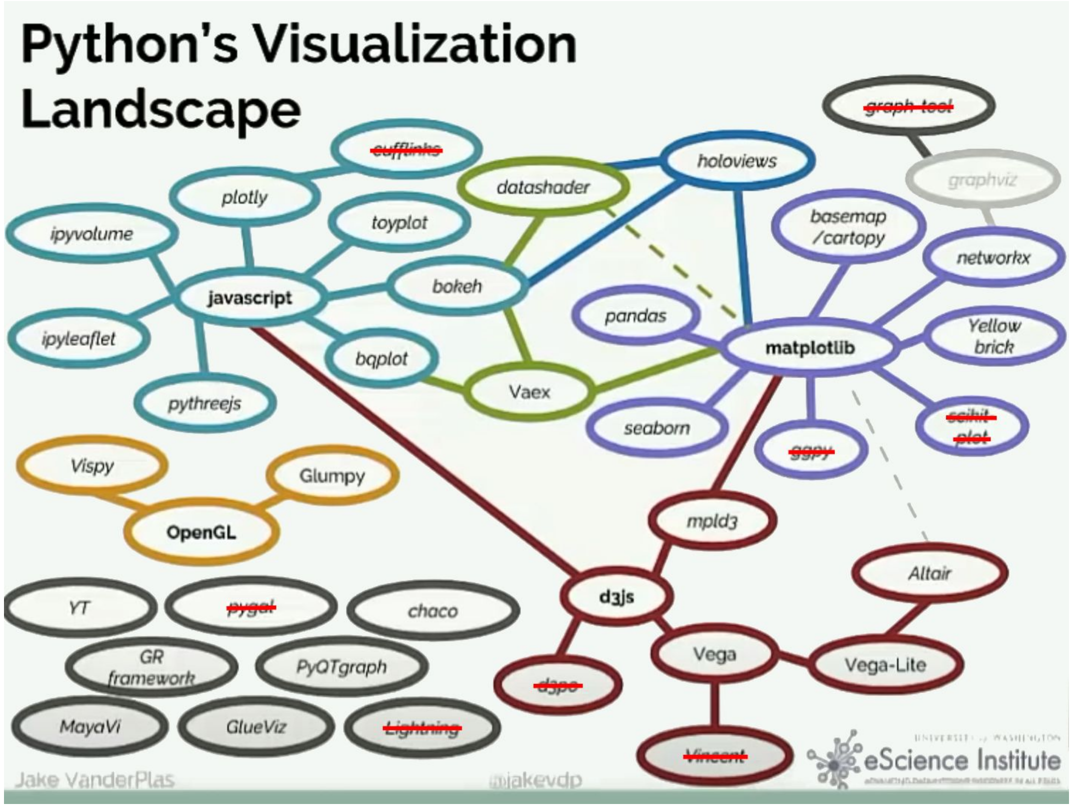
Python’s Visualization Landscape

Jake VanderPlas

@jakevdp

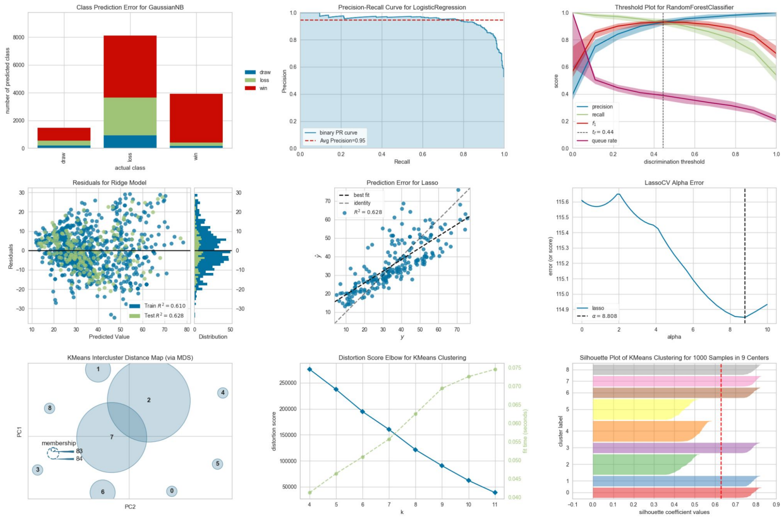
University of Washington eScience Institute

Most of these packages are still alive and well bar a few



There’s many domain specific visualisation packages

E.g. Yellowbrick which integrates scikit-learn with Matplotlib for machine learning visualisations



There’s many domains, each with many specialised visualisation packages though:

- Geospatial
- Medicine
- Meteorology
- Chemistry
- Engineering

There are too many to map or describe in one talk though - just be aware to look for packages that are specific and relevant to your problem domain as they probably exist.

Javascript Visualisation Trade Offs

⬆

Javascript visualisations are appealing since browsers are ubiquitous and Canvas, WebGL, and WebAssembly can create both complex and powerful visuals.

⬇

Interactive visuals don’t tend to scale as well.

There are few options to programmatically export images/movies from browsers (ipyvolume, ipythreejs and Plotly’s Kaleido do allow some automated exports).

Putting the Data in the Visualisation

The integrations between data packages like Pandas and XArray and visualisation packages has improved in several ways, e.g.

- Pandas added support for different data visualisation backends, e.g. Bokeh, Plotly, Altair.
- hvPlot connects multiple data sources and visualisation packages.
- New APIs like Plotly Express allow you to quickly and easily create figures from Pandas, Numpy, Geopandas, etc…

Javascript Visualisation with Python/ipywidget bindings

- ipyvizzu - Creates animated transitions between different visualisations of the same data.
- pydeck - Python bindings for deck.gl that can draw 3D maps or 3D visualisations over maps.
- pythreejs - Python bindings for Three.js - a high level library for doing WebGL rendering.
- ipycanvas - Python/ipywidget bindings for the browser canvas.
- ipycytoscape - Draws interactive network diagrams from NetworkX.

Native Visualisation Libraries with Python bindings

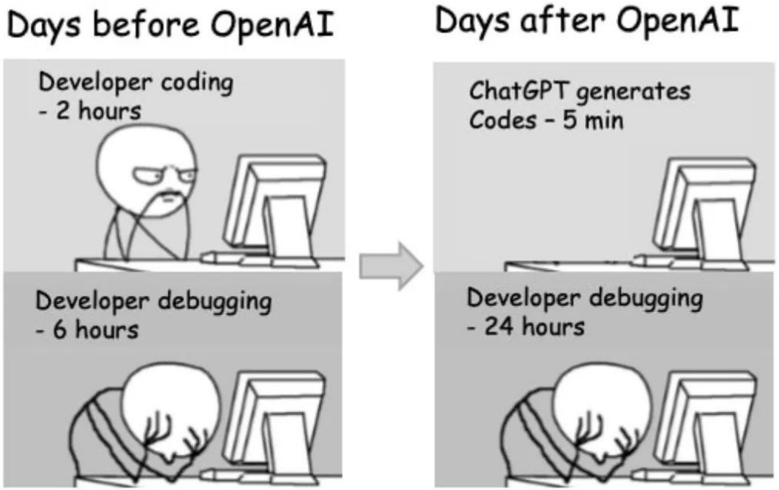
- PyVista, Vedo, Mayavi - Visualization Toolkit (VTK) based 3D data visualisation - I recommend PyVista for great documentation and community.
- Dear PyGUI - Python bindings for Dear ImGui which is a GUI API but has APIs for rendering charts and leverages native OS accelerated graphics like Metal, Vulcan, and DirectX.
- PyQtGraph - older plotting / charting library built on top of the Qt ecosystem.

Data Exploration via Visualisation

- ydata-profiling (formerly pandas-profiling) - lots of summary statistics and summary plots for Pandas dataframes.
- PyGwalker - Gives you PowerBI/Tableau like drag and drop interface to build visuals from a dataframe.
- Lux - Can analyse a dataframe and auto-generate plots that have interesting characteristics (beware it has gone a bit stale the last year).

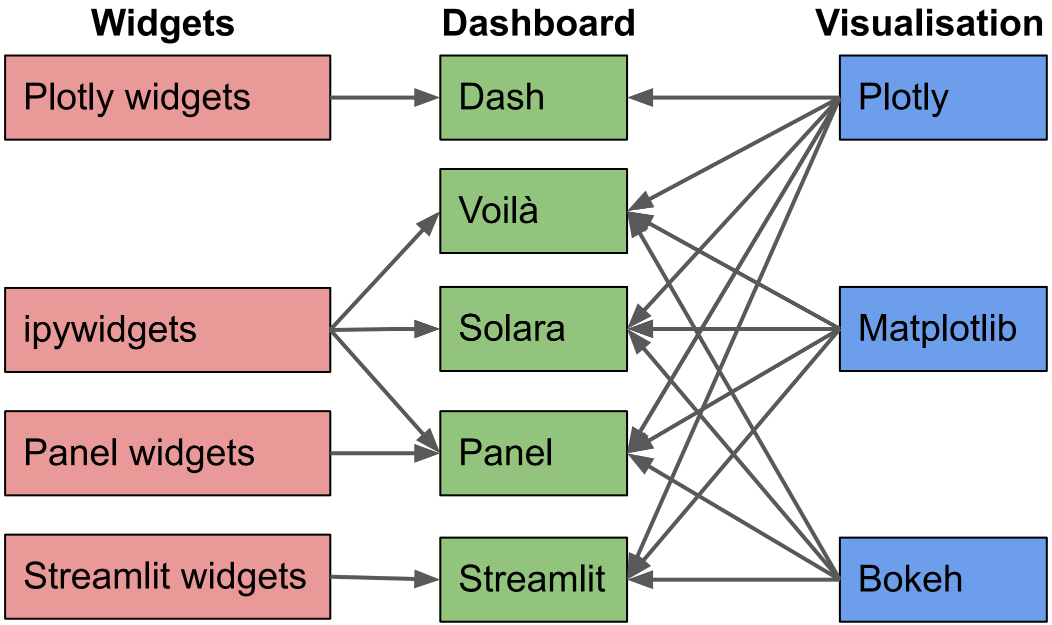
Natural Language Data Visualisation

- Sketch - Provides natural language interface to talk to a Pandas dataframe and ask questions which can include visualisations and example code as output.
- Pandas-ai - Another GPT/Pandas integration for talking to dataframes and generating visualisations (not sure it can provide the code it generates).


Days before OpenAI
Developer coding - 2 hours
Developer debugging - 6 hours
Days after OpenAI
ChatGPT generates Codes - 5 min
Developer debugging - 24 hours

Deploying Data Visuals as Dashboards

There's many competing (and cooperating) technologies for combining data visualisations with input widgets to create interactive dashboards



Widgets

Plotly widgets

ipywidgets

Panel widgets

Streamlit widgets

Dashboard

Dash

Voilà

Solara

Panel

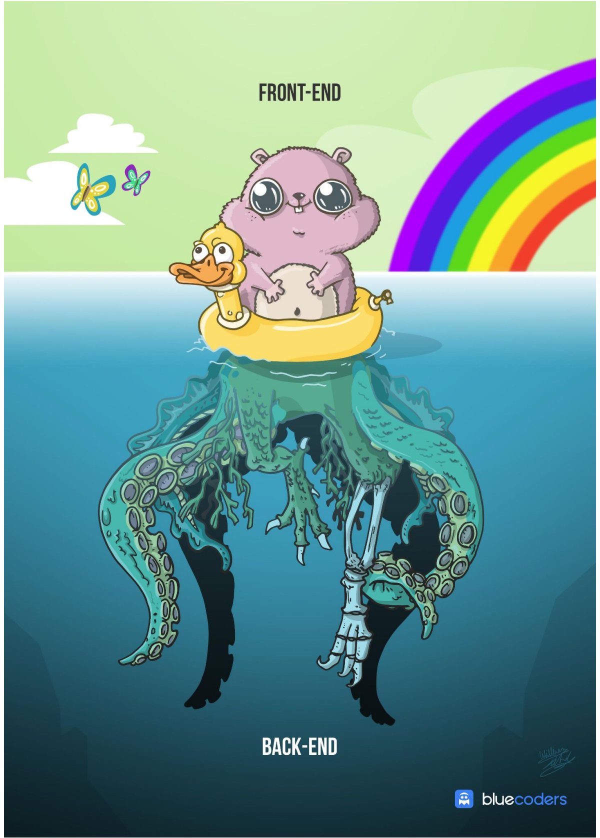
Streamlit

Visualisation

Plotly

Matplotlib

Bokeh



Python Data Visualisation inside the Browser

PyScript and Pyodide is making it possible to use staples like Matplotlib and Datashader in the browser - it also means you can use existing Javascript libraries like Bokeh / Plotly and choose the ratio of how much Python to Javascript you want.

Some of the dashboarding packages also support PyScript / pyodide like Panel and Voici (A front end version of Voilà).

Summary

- The big foundational visualisations packages in 2017 - Matplotlib, Bokeh, Plotly, and Altair are still the same in 2023 - albeit several major releases later and with tweaked APIs.
- There’s been more evolution in the visualisation and dashboarding packages built over those visualisation libraries. Panel, Voilà, and Dash have grown and Streamlit has gotten huge popularity for being a lower-code / higher abstraction alternative to them.
- I’m personally interested to see the convergence of Jupyter and web applications and more web application Python visualisations happen.

Thanks for listening!

Questions?