

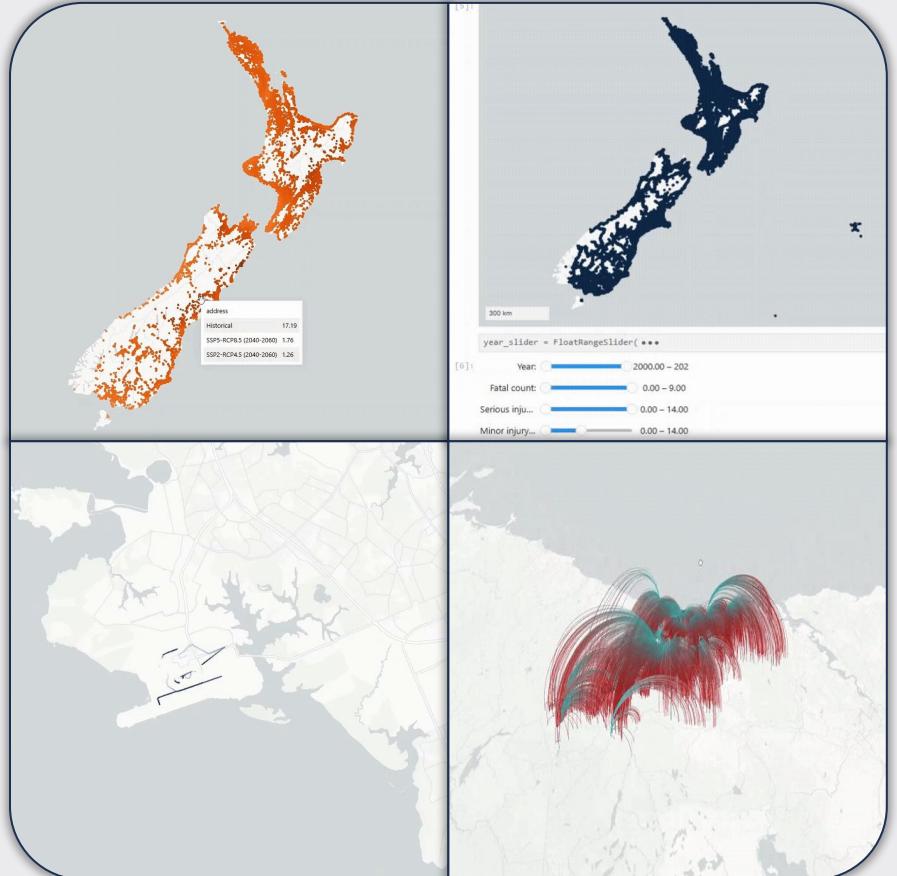
Fast and free



High-performance WebGL
geospatial visualisation in
Jupyter using Lonboard

Sam Archie

sam.archie@urbanintelligence.co.nz



• Tools, Libraries & Visualisation



Lonboard: Fast, interactive geospatial vector data visualization in Python

🕒 9:00 AM 🗓 25min 🌐 WG308 TE IRINGA

📄 Abstract

Interactive visualization is often a precursor to extracting meaningful insights from data. Lonboard provides 30-40x faster performance for visualizing geospatial vector data than other Python libraries, supporting millions of coordinates.



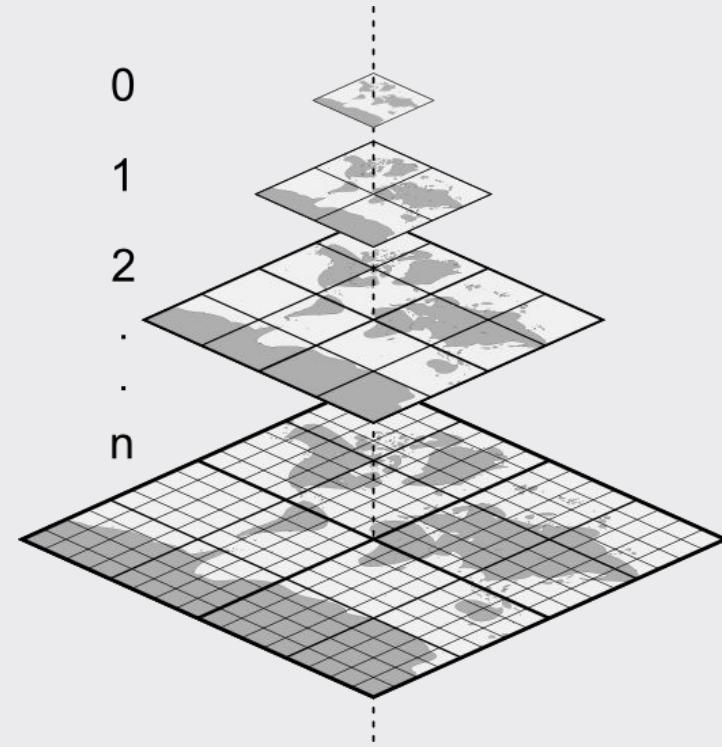
developmentSEED

Press `Esc` to close

Close



**Traditional geospatial workflows
often require expensive
preprocessing to visualise**



**How many brilliant ideas
have died because the
overhead of visualisation in
Python was too high?**



DECK.GL

GPU-powered, highly performant large-scale data visualization



geoarrow/ geoarrow-rs



GeoArrow in Rust, Python, and JavaScript
(WebAssembly) with vectorized geometry
operations

18

Contributors

64

Issues

13

Discussions

375

Stars

37

Forks



opengeospatial/ geoparquet

Specification for storing geospatial vector data
(point, line, polygon) in Parquet



26

Contributors

37

Issues

41

Discussions

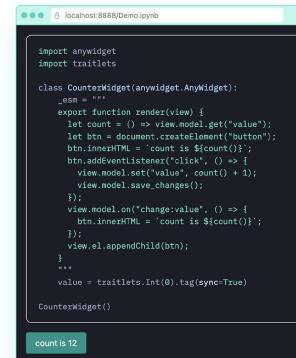
985

Stars

62

Forks

anywidget
custom jupyter widgets
made easy



```
import anywidget
import traitlets

class CounterWidget(anywidget.AnyWidget):
    _view = ''
    _view = traitlets.HTML()

    export function render(view) {
        let count = () => view.model.get("value");
        let btn = document.createElement("button");
        btn.innerHTML = "count is ${count()}" ;
        btn.addEventListener("click", () => {
            view.model.set("value", count() + 1);
            view.model.save_changes();
        });
        view.model.on("change:value", () => {
            btn.innerHTML = "count is ${count()}" ;
        });
        view_el.appendChild(btn);
    }
}

value = traitlets.Int(0).tag(sync=True)

CounterWidget()
```

count is 12



135x faster at saving data

26x smaller file sizes

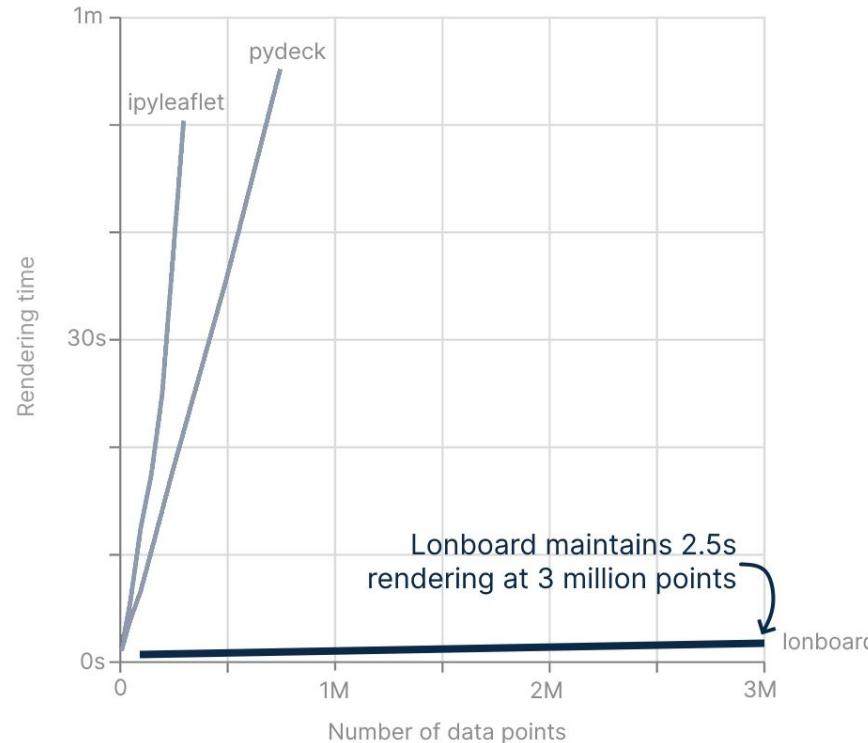
5.6x faster parsing in the browser



Performance of Lonboard vs Traditional Libraries

Time to render interactive maps with varying point densities

Adapted from <https://kylebarron.dev/blog/fast-python-bindings>



Lonboard maintains 2.5s
rendering at 3 million points

lonboard



Fast and free



High-performance WebGL
geospatial visualisation in
Jupyter using Lonboard

Sam Archie

sam.archie@urbanintelligence.co.nz



Access to the
notebooks and slides

