📄 Technical Evaluation: Plotly vs Chart.js vs D3.js for Graph Generation  
  
✅ Objective   
To identify a graphing library that:  
- Graphs very well. Great to work with.  
- Integrates smoothly with an LLM-based backend system.  
- Supports programmatic chart generation from LLM-produced JSON configs.  
- Returns URLs to interactive, hosted charts instead of rendering HTML/CSS locally.  
- Avoids dependency on frontend rendering logic or DOM manipulation.  
- Fits into a headless Python backend environment.  
  
🥇 Final Decision: Plotly + Chart Studio  
  
🔍 Comparison Table  
  
📌 Why Plotly Fits Our Use Case Best   
• Headless Backend Integration: No need for the DOM; renders fully in Python.   
• Returns Chart as Shareable URL: plotly.chart\_studio returns live chart URLs.   
• LLM JSON Compatibility: Directly feeds JSON from LLM to Plotly.   
• No HTML File Handling: Avoids local HTML/JS generation.   
• Cross-Platform Accessibility: URLs are responsive and device-agnostic.   
• Supports Multiple Chart Types: The LLM can select bar, line, pie, etc.   
• Easy Export Options: Optionally export as PNG/SVG or serve .html charts.  
  
🚫 Limitations of Chart.js / D3.js  
  
❌ Chart.js:   
- Requires DOM/canvas rendering.   
- Needs HTML and JS integration.   
- No hosted URL service.  
  
❌ D3.js:   
- Verbose, imperative code.   
- Not LLM-friendly.   
- Requires frontend-heavy rendering.   
- Poor fit for our use case.  
  
✅ Summary   
Issue: doc\_id   
Issue: issues   
Issue: guideline\_notes   
Guideline: Aim for Flesch Reading Ease ≥ 60 (higher is easier).   
Guideline: Keep average sentence length ≤ 20 words.   
Guideline: Avoid grammar and spelling mistakes.   
Guideline: Prefer active voice and clear phrasing (heuristic).