

2.1 – Tools for Creating Dashboards

(Maham Rauf)

1. Scenario – When a Dashboard Is Not the Right Tool

A dashboard would not be the right tool for visualizing the results of a qualitative survey analysis that focuses on detailed open-ended responses from participants. In such cases, the emphasis lies on themes, narratives, and textual insights rather than on quantitative metrics. Instead of a dashboard, I would use a written analytical report supported by word clouds or sentiment-analysis charts created in Python (Jupyter Notebook). This approach allows a richer interpretation of text data and helps stakeholders understand key patterns through contextual examples rather than numerical visuals.

2. Python Dashboard Analysis

Dashboard Link: [ElliotY-ML/ Dash dashboard of U.S. Covid-19 cases](#)

Main Elements:

This dashboard shows line charts and bar charts of U.S. COVID-19 cases over time, and bubble/thematic maps representing case numbers per state. Together, these components clearly communicate both geographic and temporal trends.

Visual Appeal:

It uses consistent colors to distinguish different states, clean layouts with legends and labels, and whitespace to prevent clutter. Typography and visual hierarchy guide the user's eye through trends and geographic distributions.

Informational Value:

The dashboard presents trends over time and geographical distribution side-by-side so users can compare patterns across states. It enables quick understanding of how COVID-19 spread in different regions and over time.

Interactivity & Flexibility:

Users can filter by date ranges, select states, hover to get specifics on a state, and see dynamic updates as they change filters. The interactivity gives the user control over what slice of data to examine.

3. Comparison of Python Dashboard Libraries

Library	Ease of Use	Interactivity	Scalability	Integration with Other Tools	Best For
Plotly Dash	Moderate – requires callbacks and layout structure	Excellent; supports dynamic filters, dropdowns, and live updates	High; handles large datasets efficiently	Integrates with Flask, APIs, and cloud services	Professional web dashboards and real-time analytics
Streamlit	Very easy; minimal code needed	High; offers widgets like sliders, buttons, and file uploaders	Moderate; good for small-to-medium projects	Works well with Pandas, NumPy, and scikit-learn	Rapid prototyping and portfolio apps
Panel (Bokeh)	Moderate; requires some setup	Good; supports interactive widgets and linked plots	High; handles scientific visualizations and large data well	Integrates with Bokeh, Holoviews, and Matplotlib	Scientific and exploratory dashboards
Voila (Jupyter)	Easy; converts notebooks into dashboards	Basic to moderate; limited controls	Limited; depends on notebook complexity	Seamless with Jupyter and IPython ecosystem	Teaching and simple data reports

4. Conclusion

Among the tools reviewed, Plotly Dash stands out as the best choice for developing professional, interactive, and scalable dashboards. It combines flexibility with enterprise-level features and is ideal for projects that require real-time data integration and deployment. Streamlit remains excellent for quick, visually appealing prototypes, while Panel is ideal for scientific visualizations that require precision and control.