

Ex 2.1 Tools for Creating Dashboards

Andrew Miller

1. *Think of a scenario where a dashboard would not be the right tool to visualize the results of your analysis. Create a new document (e.g., Google Docs, Pages, MS Word) and write 3 to 5 sentences describing this scenario and the tool you would use instead.*

A high school teacher giving feedback to students on their performance in a mock exam for instance. The students need specific feedback on individual topics and how they can improve. Here, aggregated metrics and graphs are not particularly useful. In this situation a written report or a spreadsheet showing a breakdown of scores would be more suitable. An individual approach like this would support meaningful learning and allow for conversations to help students understand their progress in the context of the course content. A dashboard with it's quantitative focus would struggle to achieve this.

2. *Find an example of a polished and interactive dashboard made in Python.*
3. *Paste the link to the dashboard source in your document and explain why you think this dashboard is successful. Here are the 4 questions you should answer in your analysis:*

1. *What are the main elements in the dashboard that convey the information?*
2. *What makes the dashboard visually pleasing?*
3. *What's the informational value of this dashboard?*
4. *How flexible and interactive is this dashboard from a user's perspective?*

<https://mahmoud2227.pythonanywhere.com>

In this dashboard we see various summaries of information about movies and TV series from IMDb. There are 2 or 4 graphs below a selection of buttons that change the view depending on what you want to see information about. In addition to the graphs there are 4 boxes near the top that show that there are 4532 different titles in 166 languages from 83 countries with the average number of votes being 160770. The exact meaning of the final statistic could have perhaps been clarified.

I like the colour scheme of this dashboard a lot with the yellow on the black background. The colour scheme of the graphs is highly effective using a sequential gradient scale from yellow through greens blues and into purples with yellow being the most prominent for the highest values. This is colour-blindness friendly and maintains good contrast across the entire range.

The information conveyed by the dashboard is varied and the dashboard does well to present a lot of information clearly. To avoid each view being too cluttered, the different views controlled at the top divide the visualisations up nicely so that only a manageable amount of information can be seen at any one time and the charts are well spaced out and large enough. There is a lot of informational value with the various tabs for Overview, Content creators, Parental Guide and Year and within each tab the views can be switched between Movie and Series. So overall there is a great deal of information being presented and the dashboard presents it clearly and the dashboard is easy to navigate for the user providing clear and easy access to what is wanted. There is also some practical use too with Movie and Series recommendations being offered through a further two buttons in the top right.

From a user perspective, the hover text on each graph could have been cleaned up a bit as has been done for the map but not for the other graphs.

4. *Research the existing Python libraries for dashboards and make a comparison between them. Compare a minimum of 3 Python libraries. You can use criteria such as “ease of use,” “scalability,” “integration with other tools,” and anything else that you find in your analysis to be a key factor in choosing which library to use.*

I will compare Streamlit, Plotly Dash and Panel (Holoviz)

Streamlit

The first of these, Streamlit is an open-source app framework that is easy to get started with and can be installed just like any other Python library. It has a low learning curve for analysts and is supposedly excellent for quick projects. Sliders, dropdowns, and checkboxes are intuitive and quick to add. Streamlit is good for rapid prototypes and interactive dashboards but gives less control over layout and detailed user interface features.

Streamlit performs well for small to medium teams with easy cloud deployment for internal dashboards however it is not designed for dashboards with a high number of simultaneous users. In addition, horizontal scaling, that is running the dashboard on multiple servers requires external infrastructure such as a load balancer to direct users.

In terms of integration with other tools, Streamlit integrates with Python data tools such as pandas, NumPy and Scikit-learn with easy embedding of Plotly and Matplotlib charts. It also works well in machine learning and analytics workflows, however, Streamlit has limited integration with frontend frameworks. It can also be difficult to embed Streamlit dashboards inside larger web applications.

Plotly Dash

With Dash there is more of a learning curve for new users and can be more difficult to use than Streamlit however there is more control over user interface features with highly customisable layouts and user interactions. While there is strong support for complex interactions using callbacks, this requires the analyst to have an understanding of callback logic. Dash is therefore better for complex and highly interactive dashboards once the initial learning has taken place.

In terms of scalability, Dash does support horizontal scaling with standard web infrastructure and is better suited than Streamlit for multi-user dashboards and is a strong choice for dashboards which are expected to grow in use and complexity.

Integration with Plotly's visualisation library is, of course, supported and dashboards are suitable for embedding in larger systems. Non-Plotly visualisations can be awkward and may require workarounds.

Panel (Holoviz)

Panel is a flexible and powerful tool for creating sophisticated analytical dashboards and is said to be a natural extension for users that are already familiar with Jupyter notebooks and features strong layout and interface capabilities. The learning curve is

higher than for Streamlit and comparable or possibly even steeper than for Dash. This is a good solution for advanced users who want maximal control within Python.

Scaling in terms of data complexity is well handled by Panel and large data sets are especially dealt with effectively with Datashader but scaling to many concurrent users, while possible, requires expertise and careful design.

With integration across many Python visualisation libraries, Bokeh, Matplotlib, Plotly and more can all be mixed in one dashboard and Panel is said to provide a strong notebook to dashboard workflow. Integration outside of Python is weaker and in order to fully leverage the potential of Panel, knowledge of multiple libraries is required.