

Build a Dashboard Application with Plotly Dash

In this lab, you will be building a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time.

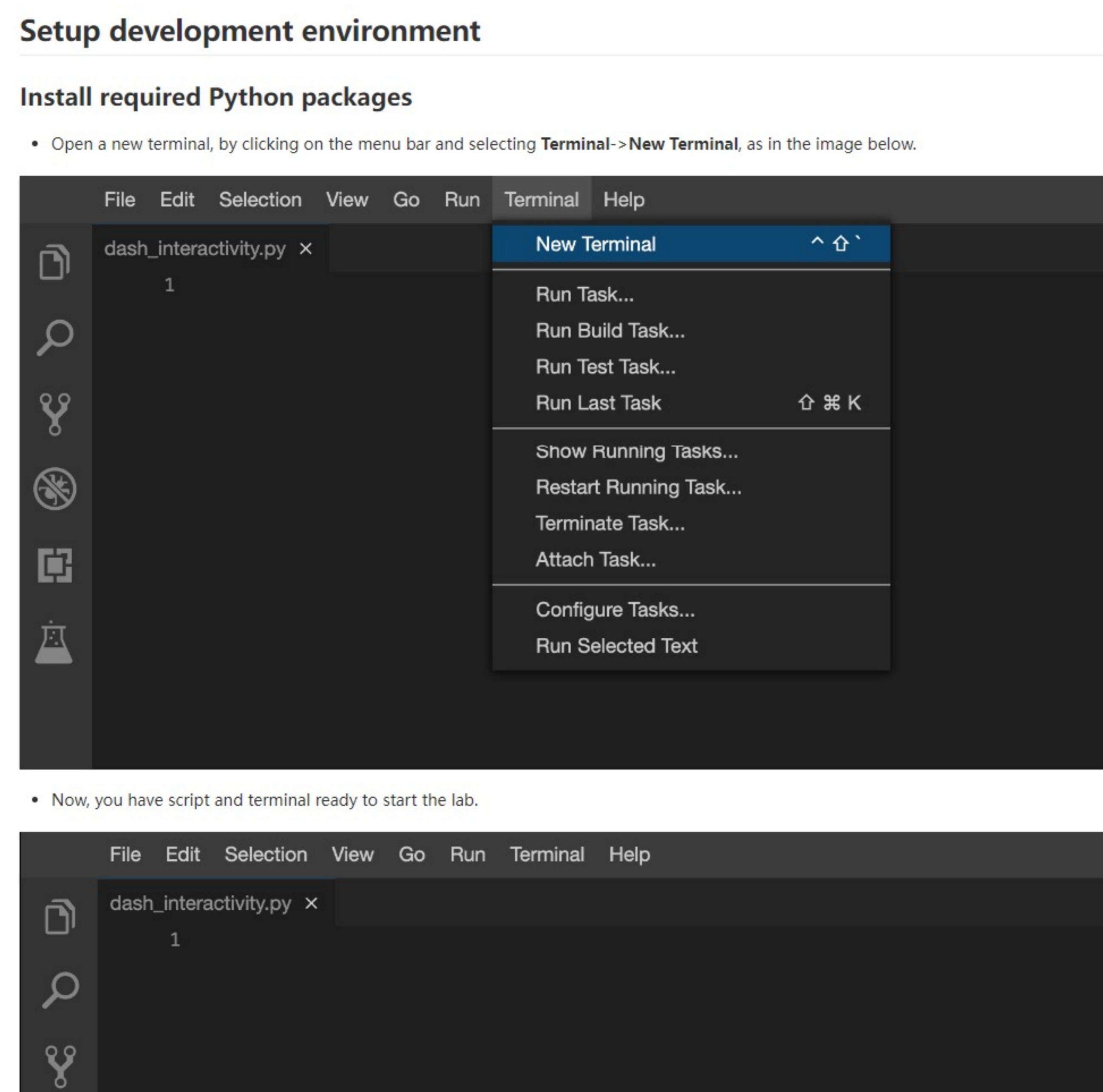
This dashboard application contains input components such as a dropdown list and a range slider. Interact with a pie chart and a scatter point chart. You will be guided to build this dashboard application via the following tasks:

- TASK 1: Add a Launch Site Drop-down Input Component
- TASK 2: Add a callback function to render `success-pie-chart`, based on selected site dropdown
- TASK 3: Add a Range Slider to Select Payload
- TASK 4: Add a callback function to render the `success-payload-scatter-chart` scatter plot

Note: Please take screenshots of the Dashboard and save them. Further upload your notebook to GitHub.

The GitHub URL and the screenshots are later required in the presentation slides.

Your completed dashboard application should look like the following screenshot:



After visual analysis using the dashboard, you should be able to obtain some insights to answer the following five questions:

- Which site has the largest successful launches?
- Which site has the highest launch success rate?
- Which payload range(s) has the highest launch success rate?
- Which payload range(s) has the lowest launch success rate?
- Which F9 Booster version (`v1.0`, `v1.1`, `FT`, `B4`, `BS`, etc.) has the highest launch success rate?

Estimated time needed: 90 minutes

Important Notice about this lab environment

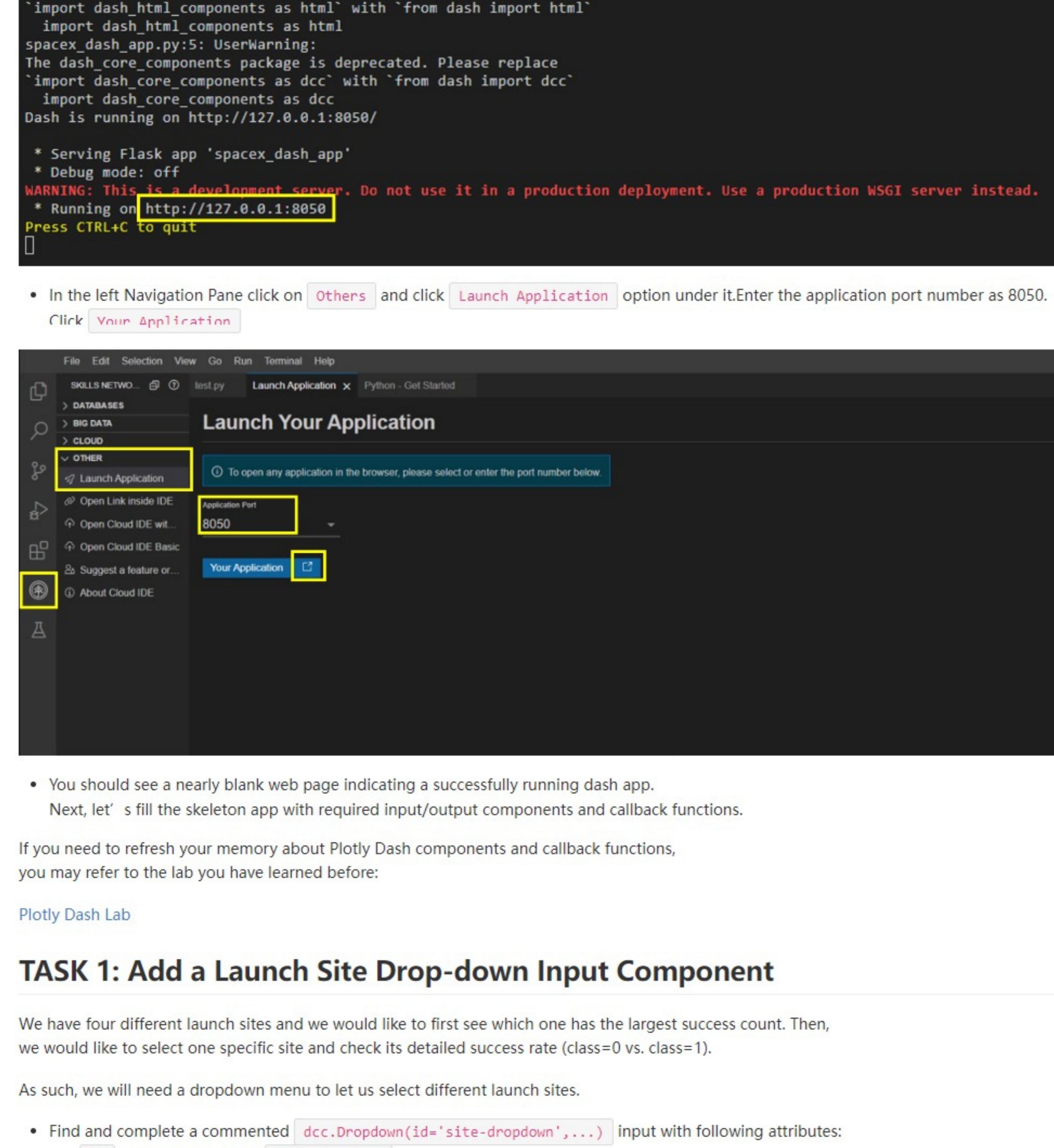
Please be aware that sessions for this lab environment are not persisted. When you launch the Cloud IDE, you are presented with a "dedicated computer on the cloud" exclusively for you. This is available to you as long as you are actively working on the labs. Once you close your session or it is timed out due to inactivity, you are logged off, and this `dedicated computer on the cloud` is deleted along with any files you may have created, downloaded or installed.

The next time you launch this lab, a new environment is created for you. If you finish only part of the lab and return later, you may have to start from the beginning. So, it is a good idea to plan your time accordingly and finish your labs in a single session.

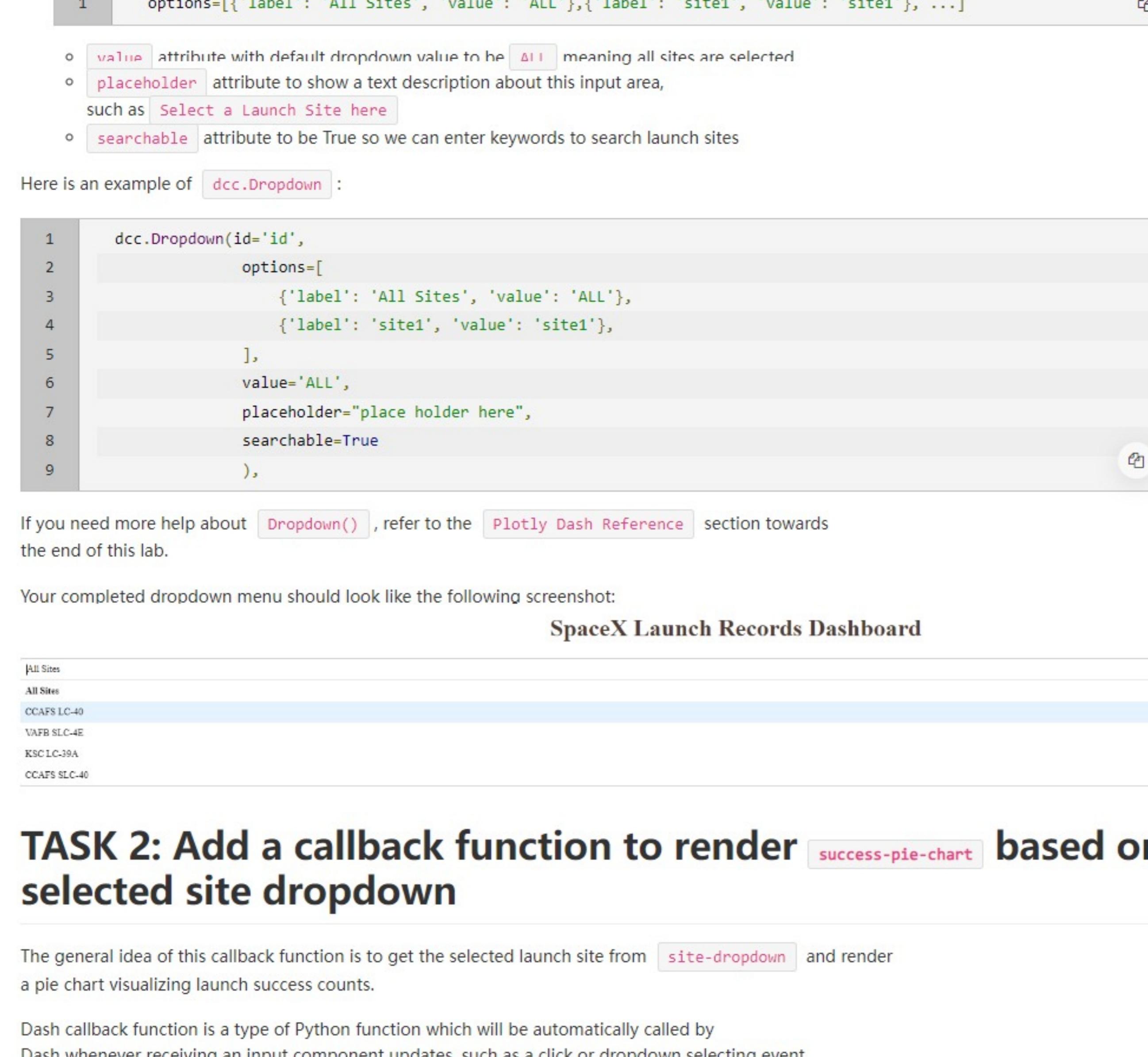
Setup development environment

Install required Python packages

Open a new terminal, by clicking on the menu bar and selecting Terminal->New Terminal, as in the image below.

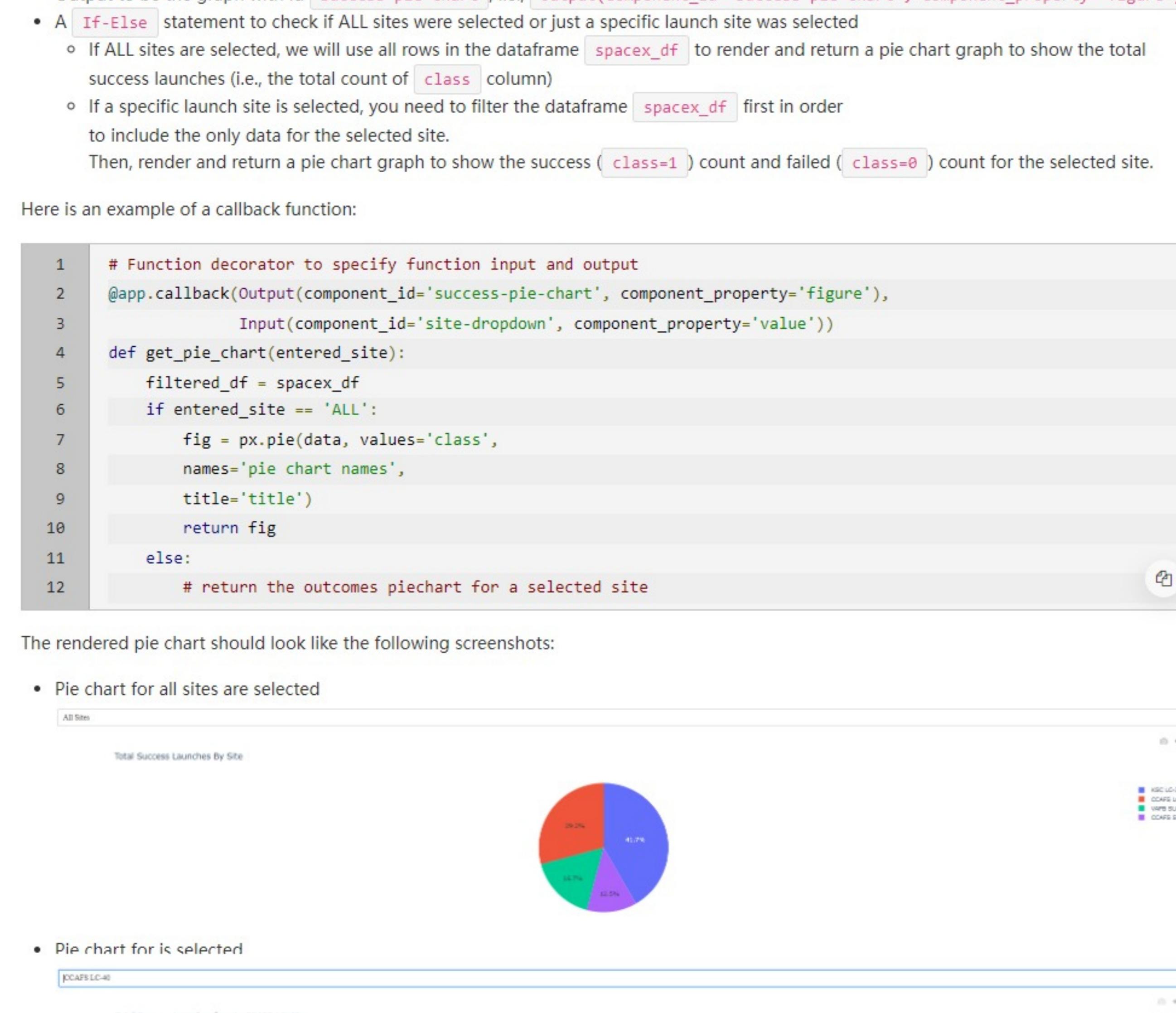


- Now, you have script and terminal ready to start the lab.

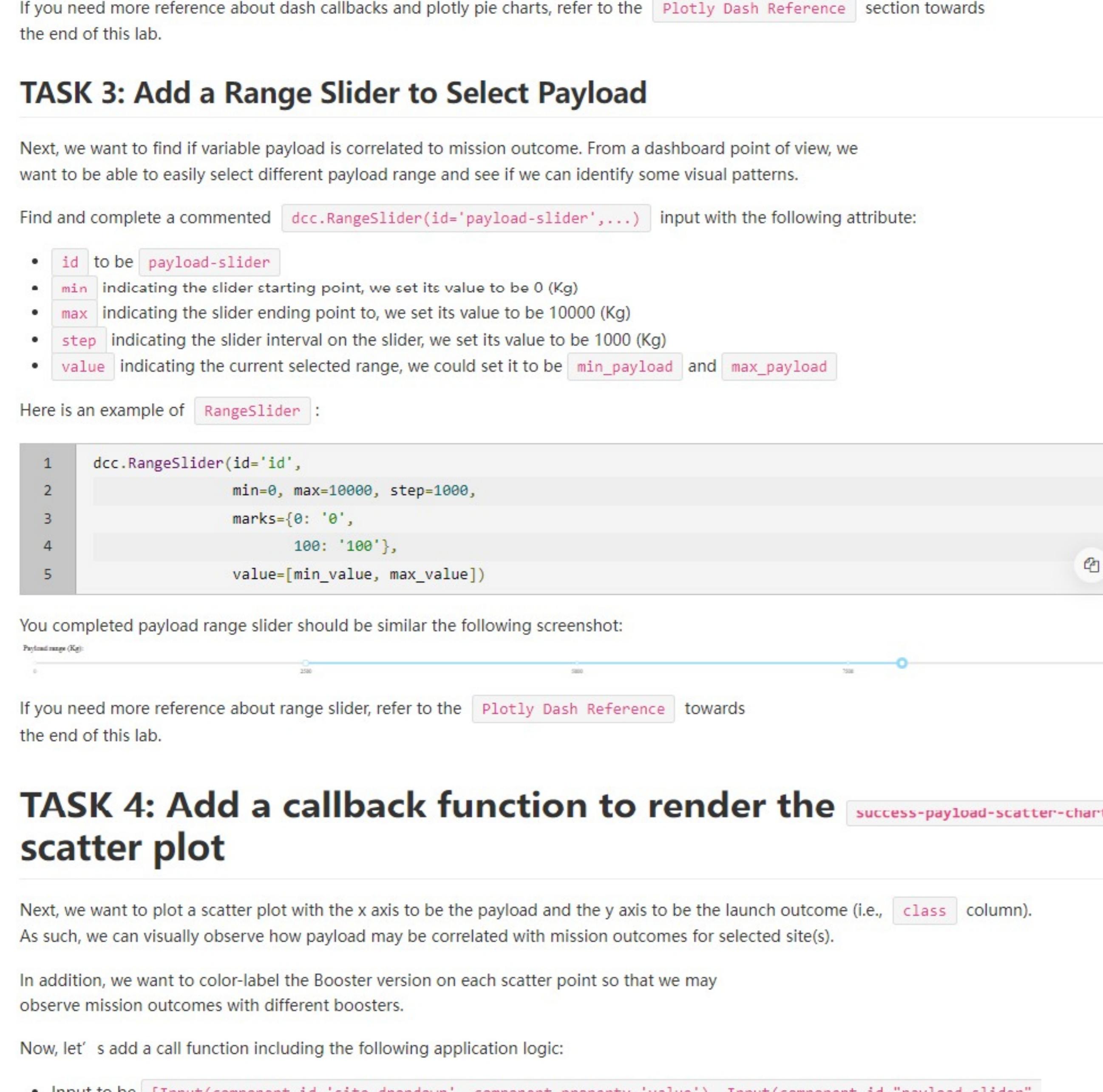


- Install python packages required to run the application.

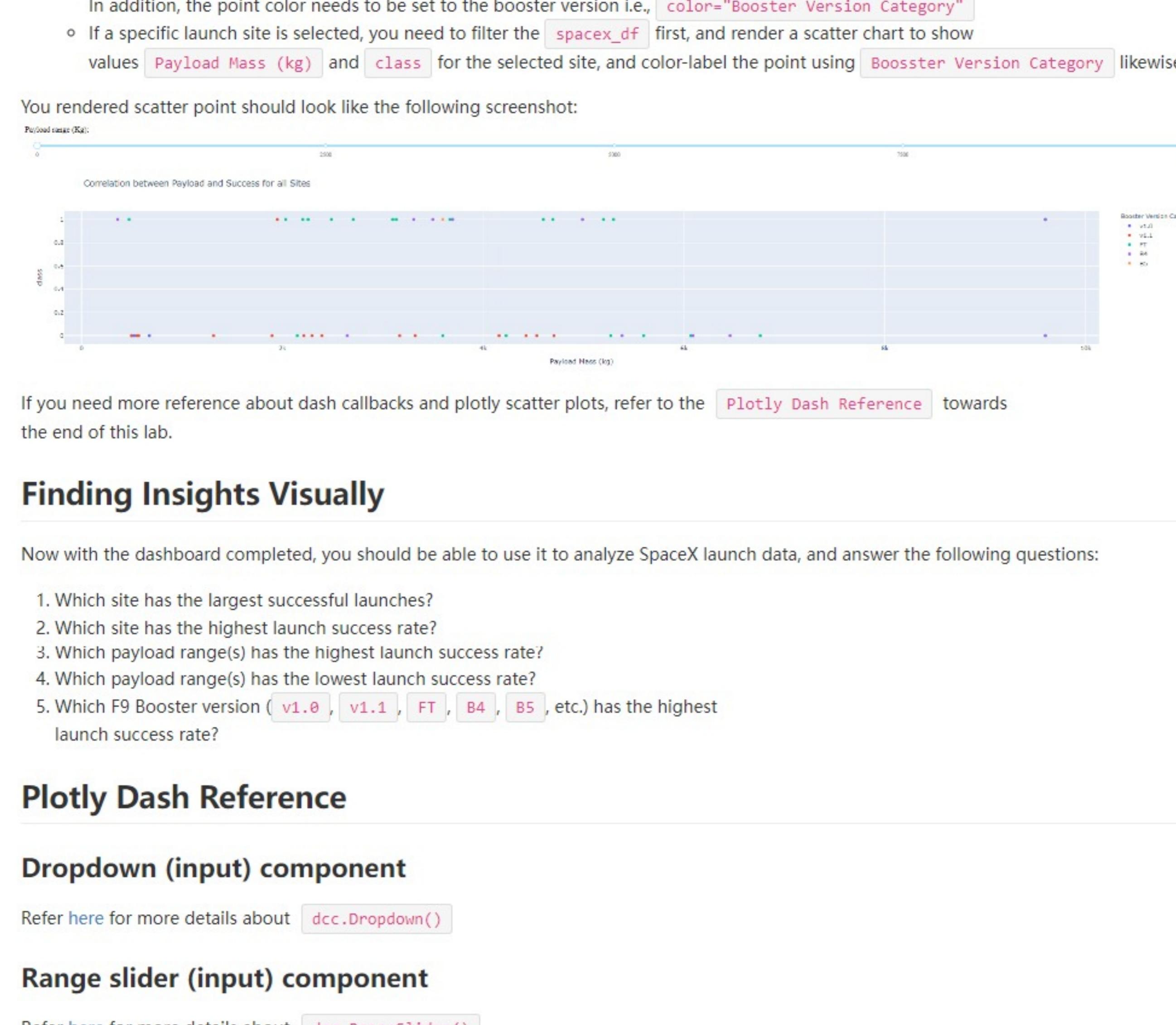
Copy and paste the below command to the terminal.



- Observe the port number (8050) shown in the terminal.



- In the left Navigation Pane click on `Others` and click `Launch Application` option under it. Enter the application port number as 8050. Click `View Application`.



- You should see a nearly blank web page indicating a successfully running dash app. Next, let's fill the skeleton app with required input/output components and callback functions.

If you need to refresh your memory about Plotly Dash components and callback functions, you may refer to the lab you have learned before:

Plotly Dash Lab

TASK 1: Add a Launch Site Drop-down Input Component

We have four different launch sites and we would like to first see which one has the largest success count. Then, we would like to select one specific launch site and check its detailed success rate (`class=0` vs. `class=1`).

As such, we will need a dropdown menu to let us select different launch sites.

- Find and complete a commented `dcc.Dropdown(id='site-dropdown',...)` input with following attributes:
 - `id` attribute with value `'site-dropdown'`
 - `options` attribute is a list of dict-like option objects (with `label` and `value` attributes). You can set the `label` and `value` all to be the launch site names in the `spacex_df` and you need to include the default `All` item, e.g.,

Here is an example of `dcc.Dropdown`:

If you need more help about `Dropdown()`, refer to the `Plotly Dash Reference` section towards the end of this lab.

Your completed dropdown menu should look like the following screenshot:

- Pie chart for all sites are selected
- Pie chart for is selected

- If you need more reference about dash callbacks and plotly pie charts, refer to the `Plotly Dash Reference` section towards the end of this lab.

Finding Insights Visually

Now with the dashboard completed, you should be able to use it to analyze SpaceX launch data, and answer the following questions:

- Which site has the largest successful launches?
- Which site has the highest launch success rate?
- Which payload range(s) has the highest launch success rate?
- Which payload range(s) has the lowest launch success rate?

5. Which F9 Booster version (`v1.0`, `v1.1`, `FT`, `B4`, `BS`, etc.) has the highest launch success rate?

Plotly Dash Reference

Dropdown (input) component

Refer here for more details about `dcc.Dropdown()`.

Range slider (input) component

Refer here for more details about `dcc.RangeSlider()`.

Pie chart (output) component

Refer here for more details about `plotly pie charts`.

Scatter chart (output) component

Refer here for more details about `plotly scatter charts`.

Author

Yan Luo

Other contributor(s)

Joseph Santarcangelo