Plotly graphs and figures

BUILDING DASHBOARDS WITH DASH AND PLOTLY

Alex Scriven
Data Scientist

datacamp

What is Dash?

- A Python library for creating interactive web applications

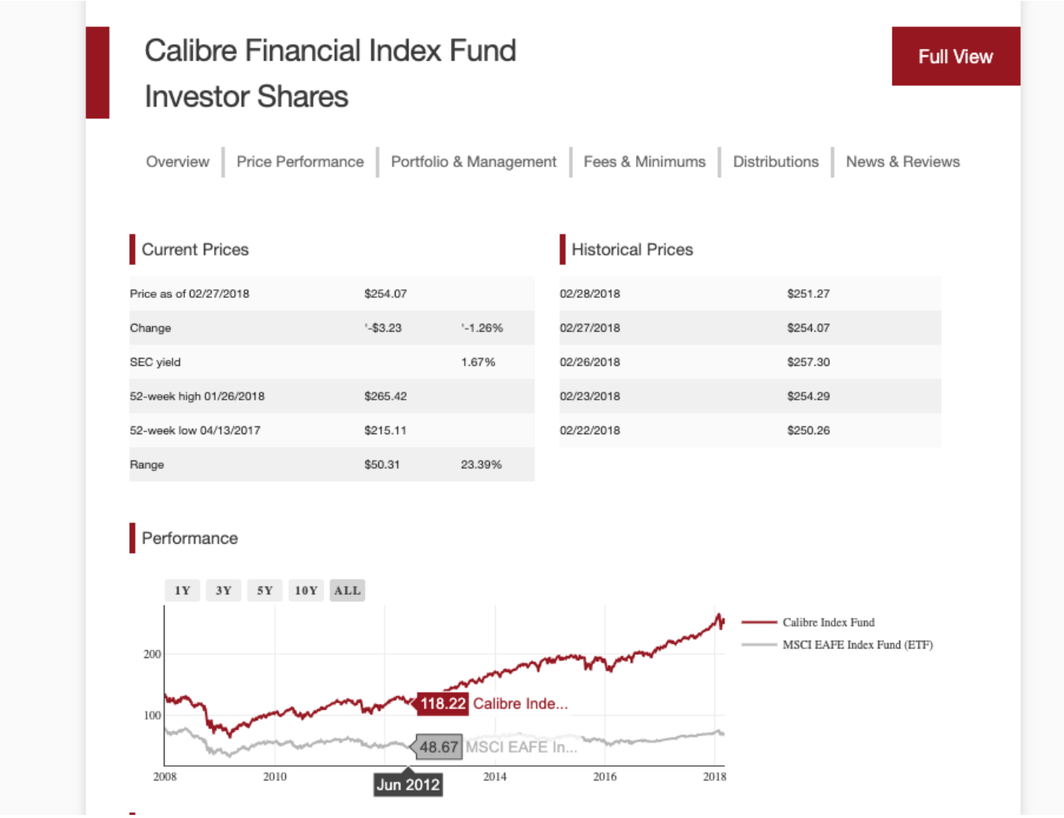
Advantages:

- Free! Unlike Tableau or PowerBI
- Harness JavaScript with only Python
- Less code than web application frameworks

Plotly and Dash

- Work well together

- Dash: Interactive dashboards with multiple Plotly graphs
- See this **example**
  - Images, text and charts
  - Check out the **source code**



What is Plotly?

- Refresh Plotly, focus on Dash
- A Python library for creating modern, interactive graphs
  - Wraps JavaScript but code in Python
- Create graphs with `plotly.express`

- Check out the **Introduction to Data Visualization with Plotly in Python**

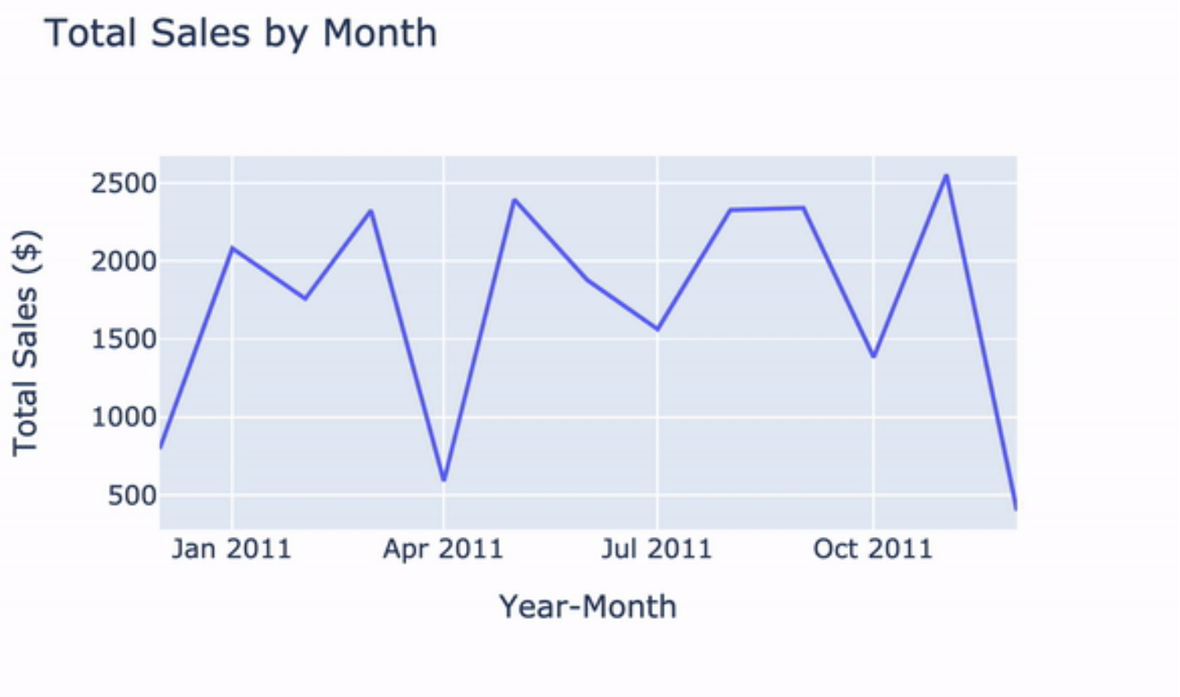
Our e-commerce data

- Dataset of e-commerce sales

- Details:
  - Item category (Major, Minor) + description
  - Unit price, quantity (+ OrderValue)
  - Country
  - Year-Month of sale

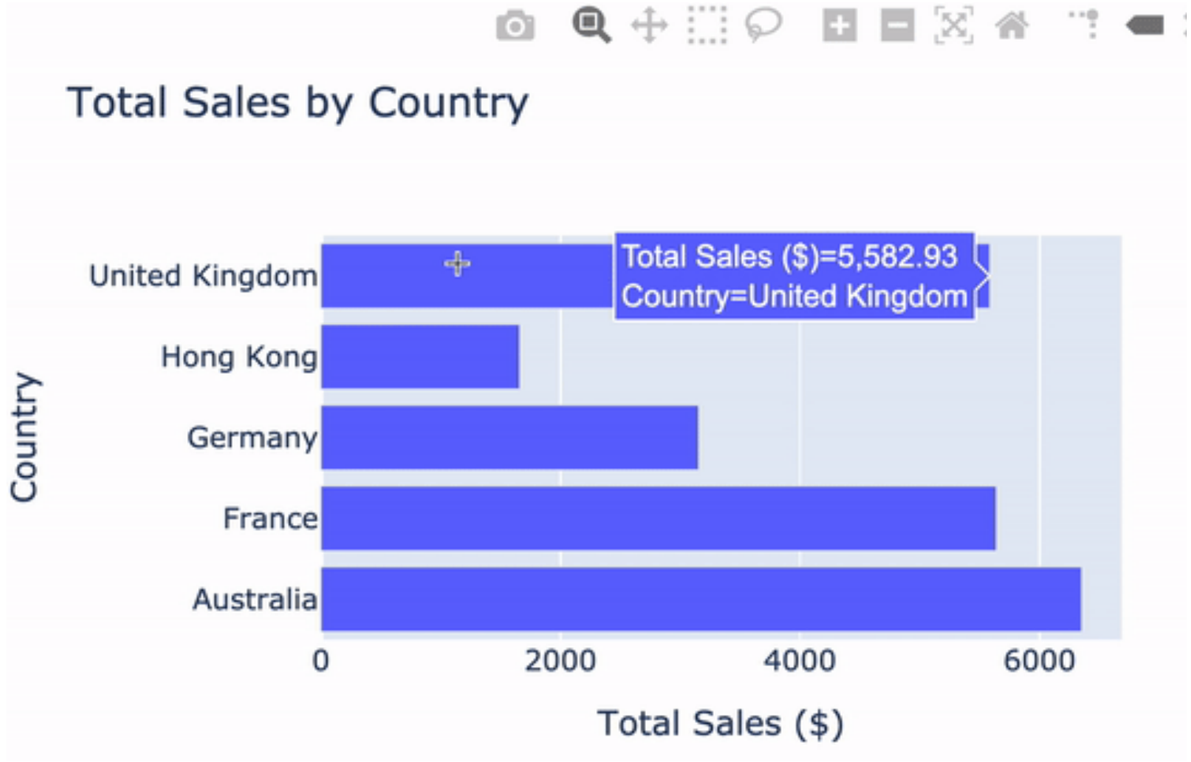
Line charts with plotly.express

```
import plotly.express as px
line_graph = px.line(
  data_frame=ecom_sales,
  x='Year-Month',
  y='Total Sales ($)',
  title='Total Sales by Month')
line_graph.show()
```



Bar charts with plotly.express

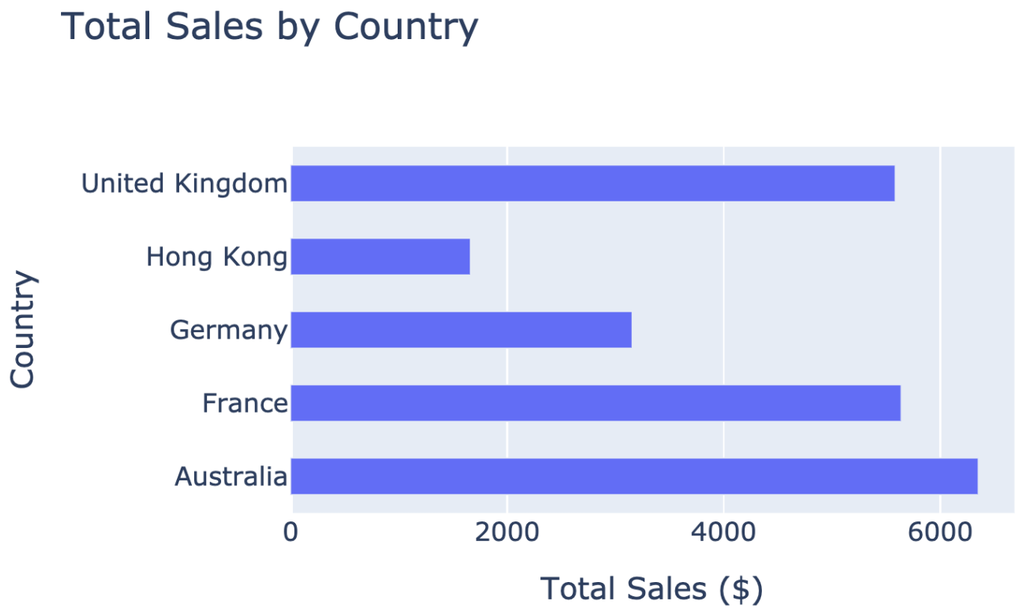
```
bar_fig = px.bar(
  data_frame=ecom_sales,
  x='Total Sales ($)',
  y='Country',
  title='Total Sales by Country',
  orientation='h')
bar_fig.show()
```



Customizing Plotly graphs

Changing the bar width of our bar graph:

```
bar_fig.update_layout({'bargap': 0.5})
bar_fig.show()
```



Check out the **Plotly documentation**

Let's practice!

BUILDING DASHBOARDS WITH DASH AND PLOTLY

From Plotly to Dash

BUILDING DASHBOARDS WITH DASH AND PLOTLY

Alex Scriven
Data Scientist

datacamp

A first Dash App

A complete Dash app:

```
from dash import Dash, dcc
app = Dash()
app.layout = [dcc.Graph(id='example-graph', figure=bar_fig)]
if __name__ == '__main__':
    app.run(debug=True)
```

- Python functionality possible
  - e.g., String interpolation `print("f{my_variable}")`

The main Dash imports

```
from dash import Dash, dcc
```

- `Dash` initializes the app
- `dcc` has UI components like graphs and inputs
  - One component in our app
  - More components throughout the course

The app layout

```
app = Dash()
app.layout = [dcc.Graph(
  id='example-graph',
  figure=bar_fig)]
```

- Create an app object using `Dash()`
- Set the `app.layout`
  - Using `dcc.Graph()`
    - `figure` = The Plotly figure to render
    - `id` = Important for callbacks later

Running the app

```
if __name__ == '__main__':
    app.run(debug=True)
```

- Script is run from command-line
  - i.e., `python my_app.py` in the command-line
- `debug` for feedback when testing

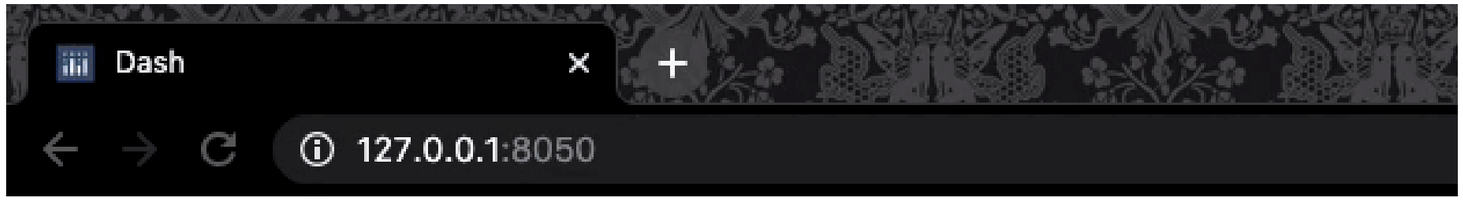
Our app

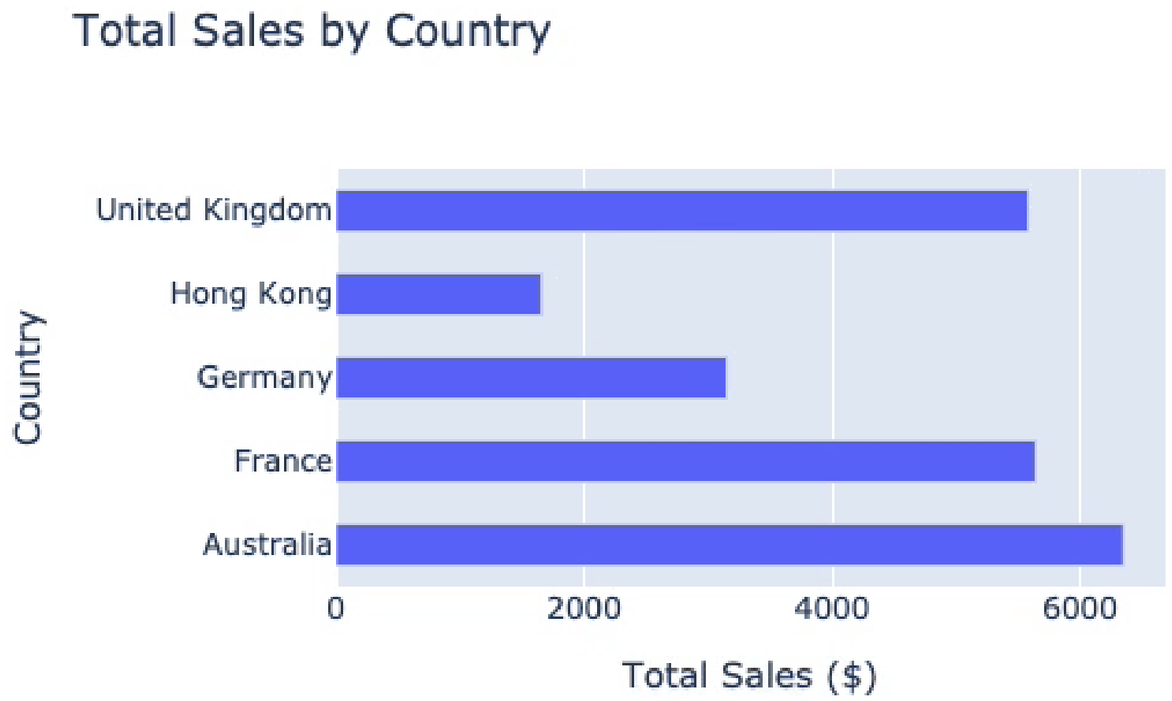
- Access via a web browser such as Google Chrome
- While served, update and save `.py` file to see live updates in browser

```
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "simple_app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
```

Our app in the browser





Dash in DataCamp

- Some differences to other DataCamp exercises:
  - Setup lines at the top of the exercises.
  - All executed at once (not line-by-line)
  - (Much) longer code
- Fully-functional dashboards (expand the window to see!)

Let's practice!

BUILDING DASHBOARDS WITH DASH AND PLOTLY

Positioning Dash components

BUILDING DASHBOARDS WITH DASH AND PLOTLY

Alex Scriven
Data Scientist

datacamp

HTML and the web

HTML: language for structuring websites

- HTML: wooden structure of a house
  - Set placement of objects
- CSS: paint color of a room
  - Style (e.g., background color) of objects

Div and H tags

- Dash uses `dash.html` to interface between HTML and Python

Two important HTML structures ('tags'):

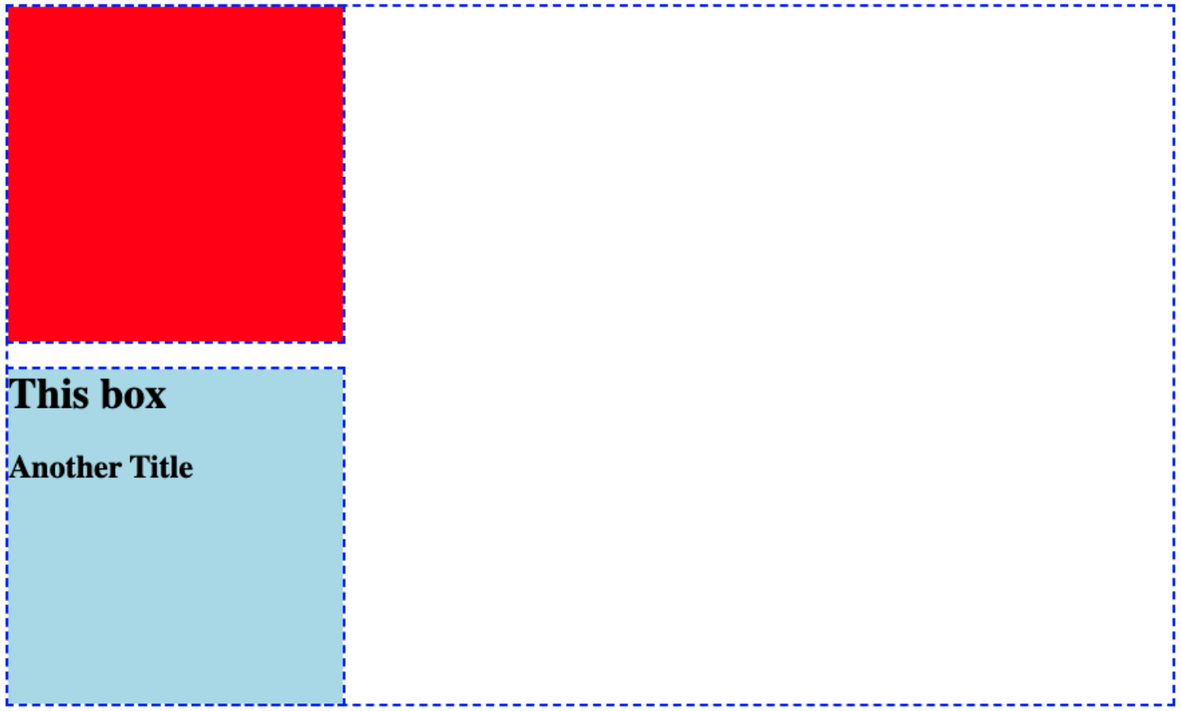
- Div tags: important for structuring websites
  - Can have many different-sized divs with different things inside
- H tags: different sized titles (H1 > H6)

Using Div and H tags

- Overall div (everything inside)
- Div inside: red background
- Div with blue background
  - H tags inside
- Ignore the `style` part - more on 'CSS' later!

```
<div>
  <div style="background-color: red;
              width:250; height:250;">
  </div>
  <div style="background-color: lightblue;
              width:250; height:250;">
    <h1>This box</h1>
    <h2>Another Title</h2>
  </div>
</div>
```

Our example displayed



Take note:

- Red background div
- Blue background div with H tags

Our example in Dash

- Recreating HTML example with Dash

```
from dash import Dash, html
app = Dash()
app.layout = [
    html.Div(style={'height':250, 'width':250, 'background-color':'red'}),
    html.Div(children=[
        html.H1("This box"),
        html.H2("Another Title")],
        style={'background-color':'lightblue'})
    ]
```

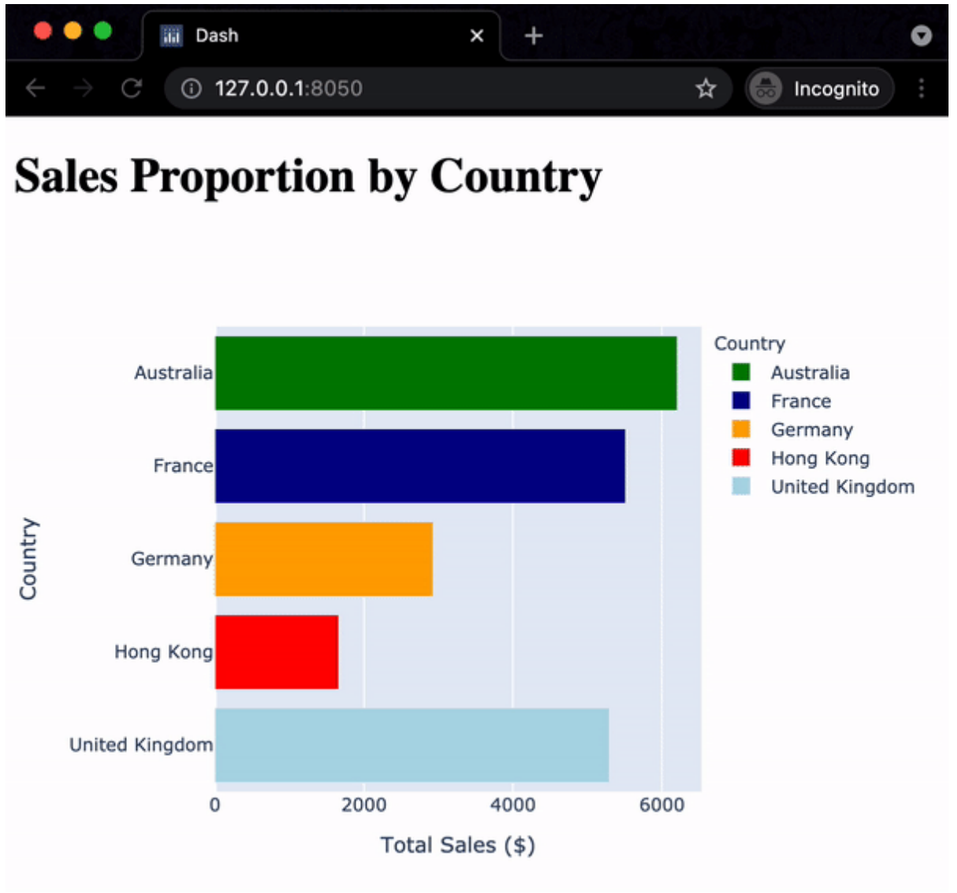
Breaking down the layout

- HTML tags align to Dash `html.()`
  - `html.Div()` = `<div>`
  - `html.H1()` = `<h1>`
- The last div has a `children` argument
  - A list of objects to go inside
- We can put `dcc.Graph()` components inside too!

```
from dash import Dash, html
app.layout = [
  html.Div(
    style={'background-color':'red',
           'height':250, 'width':250}),
  html.Div(
  children=[
      html.H1("This box"),
      html.H2("Another Title")]
    ,style={'background-color':'lightblue',
              'height':250, 'width':250})
]
```

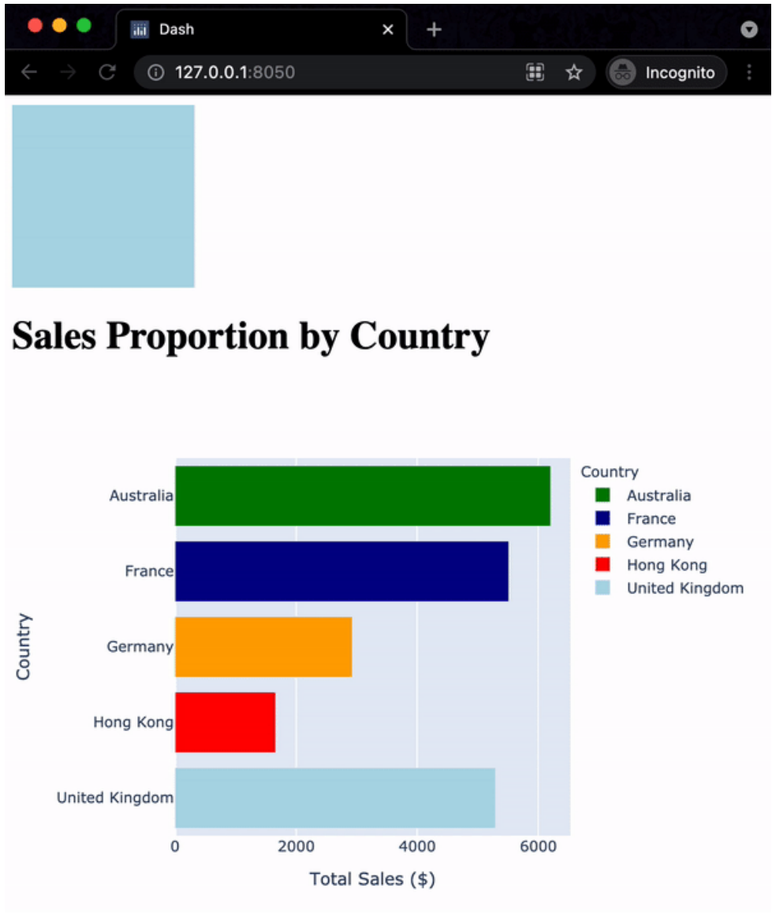
Graphs in the layout

```
bar_fig_country = px.bar(ecom_sales,
 x='Total Sales ($)', y='Country',
 color='Country', color_discrete_map=
 {'United Kingdom':'lightblue',
 'Germany':'orange', 'France':'darkblue',
 'Australia':'green', 'Hong Kong':'red'})
app = Dash()
app.layout = [
  html.H1("Sales Proportion by Country"),
  dcc.Graph(id='bar_graph',
            figure=bar_fig_country)
]
```



Adding more structure

```
app.layout = [
  html.Div(style={
    'width':150,'height':150,
    'background-color':'lightblue'}),
  html.H1("Sales Proportion by Country"),
  dcc.Graph(id='bar_graph',
            figure=bar_fig_country)
]
```



Let's practice!

BUILDING DASHBOARDS WITH DASH AND PLOTLY