**Let's start creating dash application**

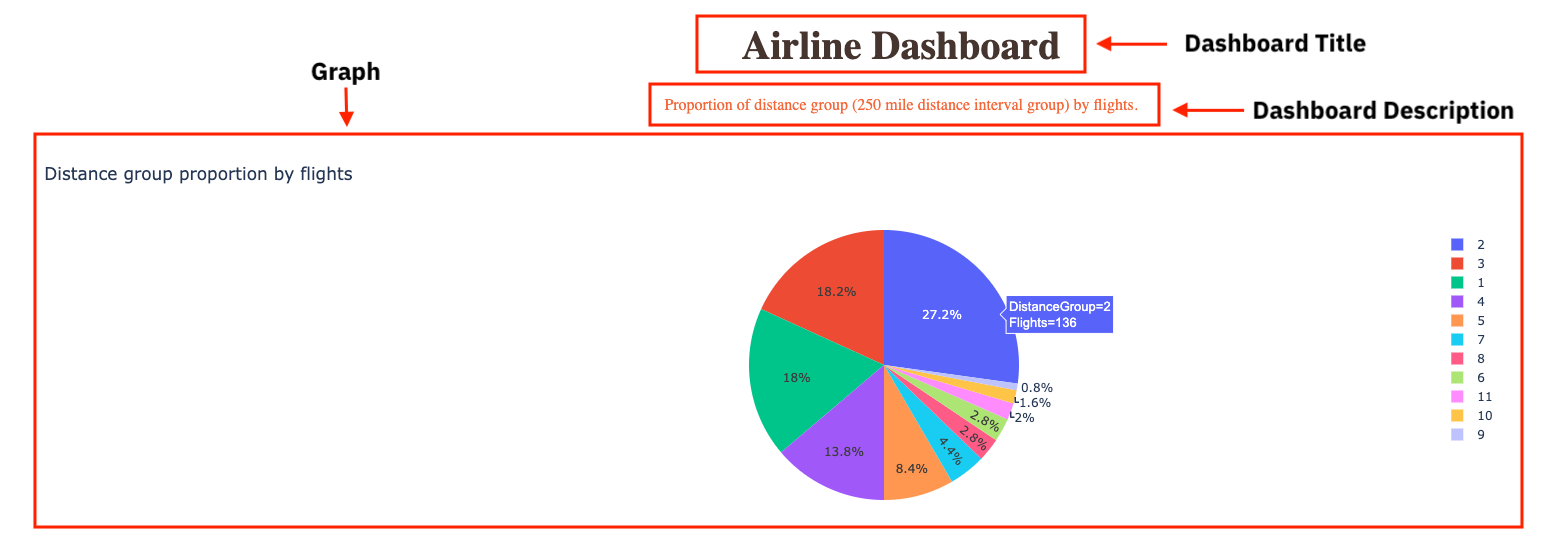
**Goal**

Create a dashboard that displays the percentage of flights running under specific distance group. Distance group is the distance intervals, every 250 miles, for flight segment. If the flight covers to 500 miles, it will be under distance group 2 (250 miles + 250 miles).

**Expected Output**

Below is the expected result from the lab. Our dashboard application consists of three components:

* Title of the application
* Description of the application
* Chart conveying the proportion of distance group by month

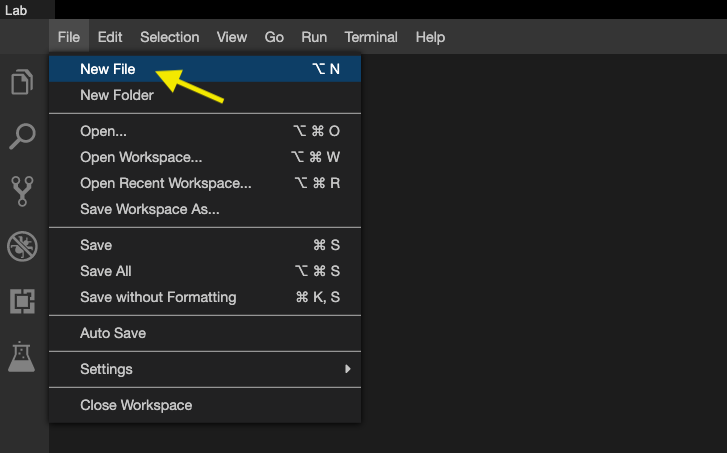


**To do:**

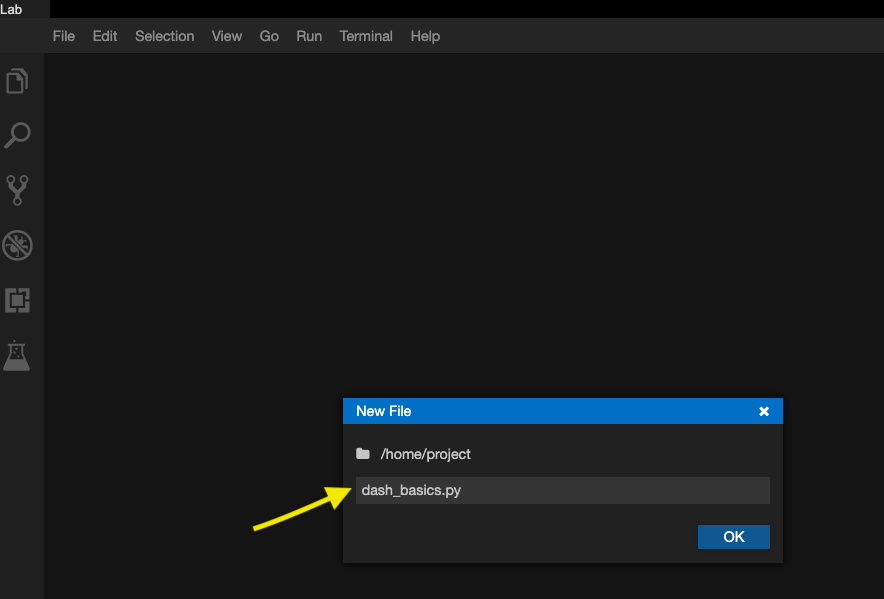
1. Import required libraries and read the dataset
2. Create an application layout
3. Add title to the dashboard using HTML H1 component
4. Add a paragraph about the chart using HTML P component
5. Add the pie chart above using core graph component
6. Run the app

**Get the tool ready**

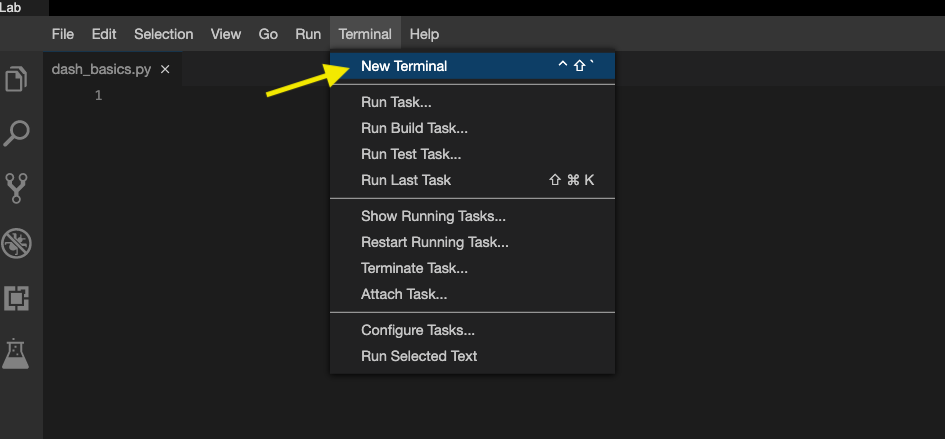
* Create a new python script, by clicking on the menu bar and selecting **File**->**New File**, as in the image below.



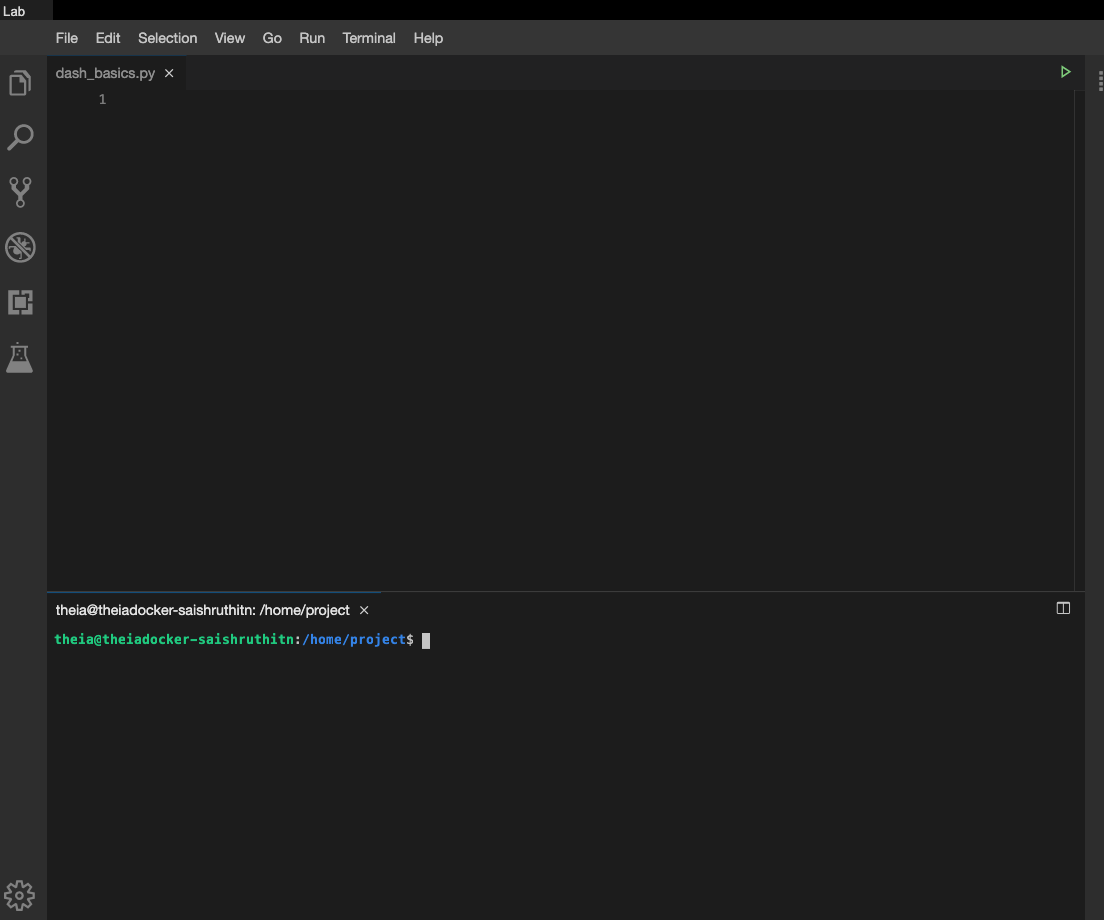
* Provide the file name as dash\_basics.py



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

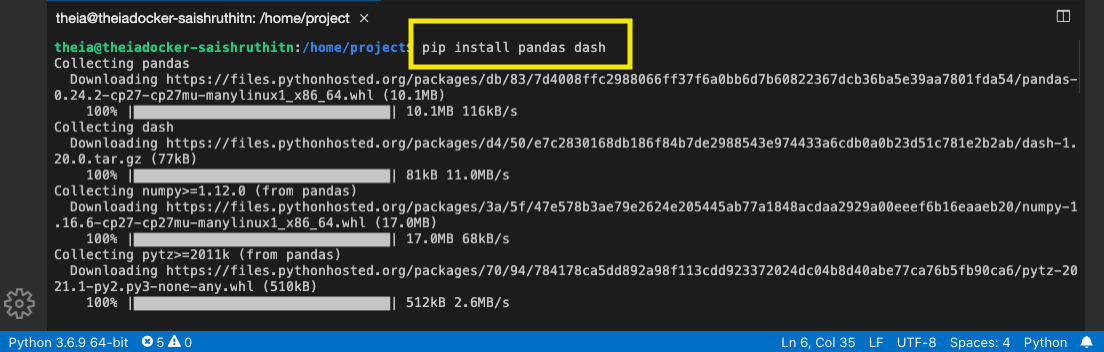


* Now, you have script and terminal ready to start the lab.



* Install python packages required to run the application. Copy and paste the below command to the terminal.

**pip3** install pandas dash



**TASK 1 - Data Preparation**

Let's start with

* Importing necessary libraries
* Reading and sampling 500 random data points
* Get the chart ready

Copy the below code to the dash\_basics.py script and review the code.

# Import required packages

**import** pandas **as** pd

**import** plotly.express **as** px

**import** dash

**import** dash\_html\_components **as** html

**import** dash\_core\_components **as** dcc

# Read the airline data into pandas dataframe

airline\_data = pd.read\_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DV0101EN-SkillsNetwork/Data%20Files/airline\_data.csv',

encoding = "ISO-8859-1",

dtype={'Div1Airport': str, 'Div1TailNum': str,

'Div2Airport': str, 'Div2TailNum': str})

# Randomly sample 500 data points. Setting the random state to be 42 so that we get same result.

data = airline\_data.sample(n=500, random\_state=42)

# Pie Chart Creation

fig = px.pie(data, values='Flights', names='DistanceGroup', title='Distance group proportion by flights')

**TASK 2 - Create dash application and get the layout skeleton**

Next, we create a skeleton for our dash application. Our dashboard application has three components as seen before:

* Title of the application
* Description of the application
* Chart conveying the proportion of distance group by month

Mapping to the respective Dash HTML tags:

* Title added using html.H1() tag
* Description added using html.P() tag
* Chart added using dcc.Graph() tag

Copy the below code to the dash\_basics.py script and review the structure.

*NOTE*: Copy below the current code

# Create a dash application

**app** = dash.Dash(\_\_name\_\_)

# Get the layout of the application and adjust it.

# Create an outer division using html.Div and add title to the dashboard using html.H1 component

# Add description about the graph using HTML P (paragraph) component

# Finally, add graph component.

app.layout = html.Div(children=[html.H1(),

html.P(),

dcc.Graph(),

])

# Run the application

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server()

**TASK 3 - Add the application title**

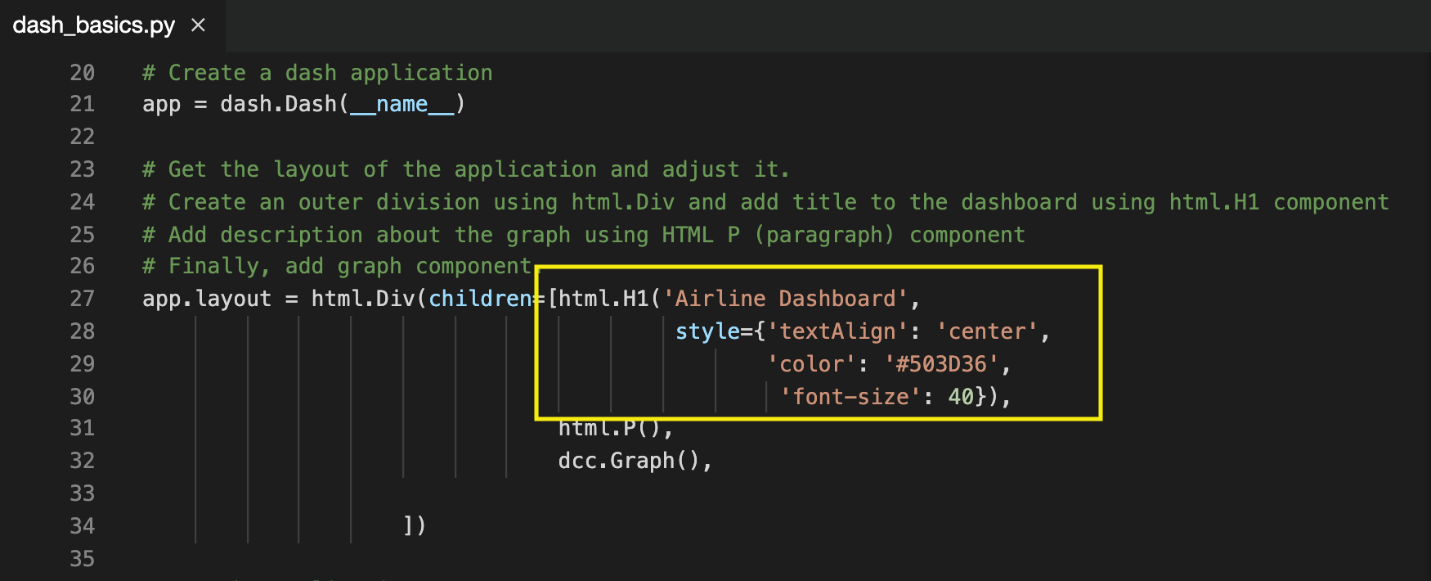
Update the html.H1() tag to hold the application title.

* Application title is Airline Dashboard
* Use style parameter provided below to make the title center aligned, with color code #503D36, and font-size as 40

'Airline Dashboard',

style={'textAlign': 'center', 'color': '#503D36', 'font-size': 40}

After updating the html.H1() with the application title, the app.layout will look like:



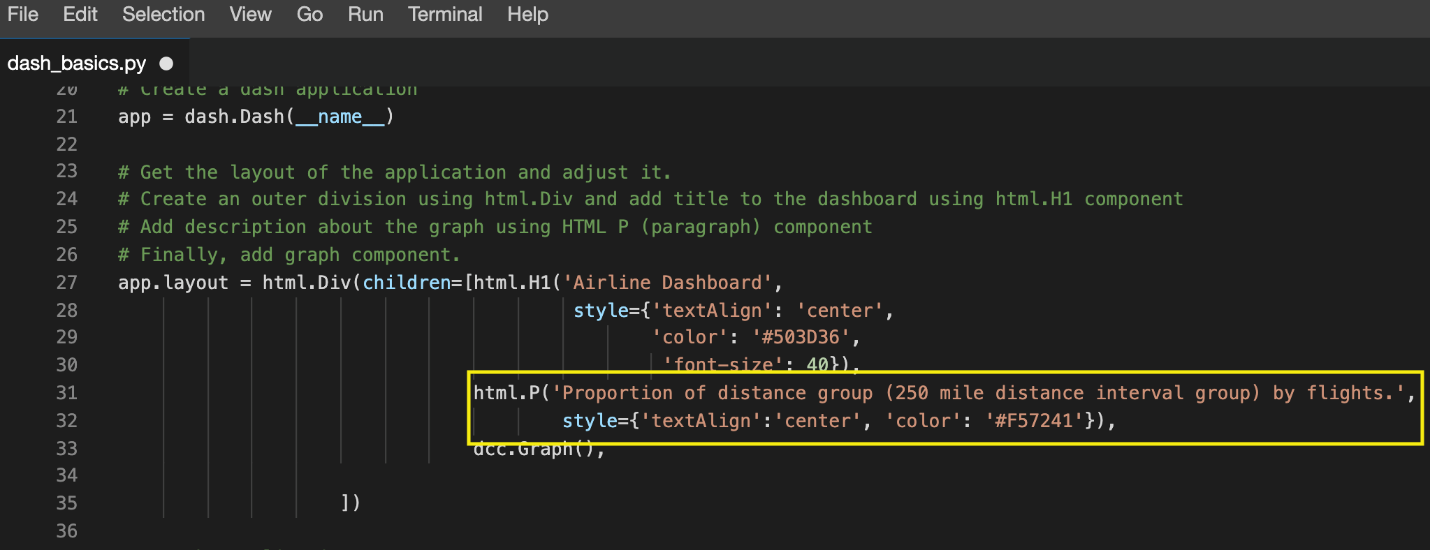
**TASK 4 - Add the application description**

Update the html.P() tag to hold the description of the application.

* Description is Proportion of distance group (250 mile distance interval group) by flights.
* Use style parameter to make the description center aligned and with color #F57241.

html.P('Proportion of distance group (250 mile distance interval group) by flights.', style={'textAlign':'center', 'color': '#F57241'}),

After updating the html.H1() with the application title, the app.layout will look like:

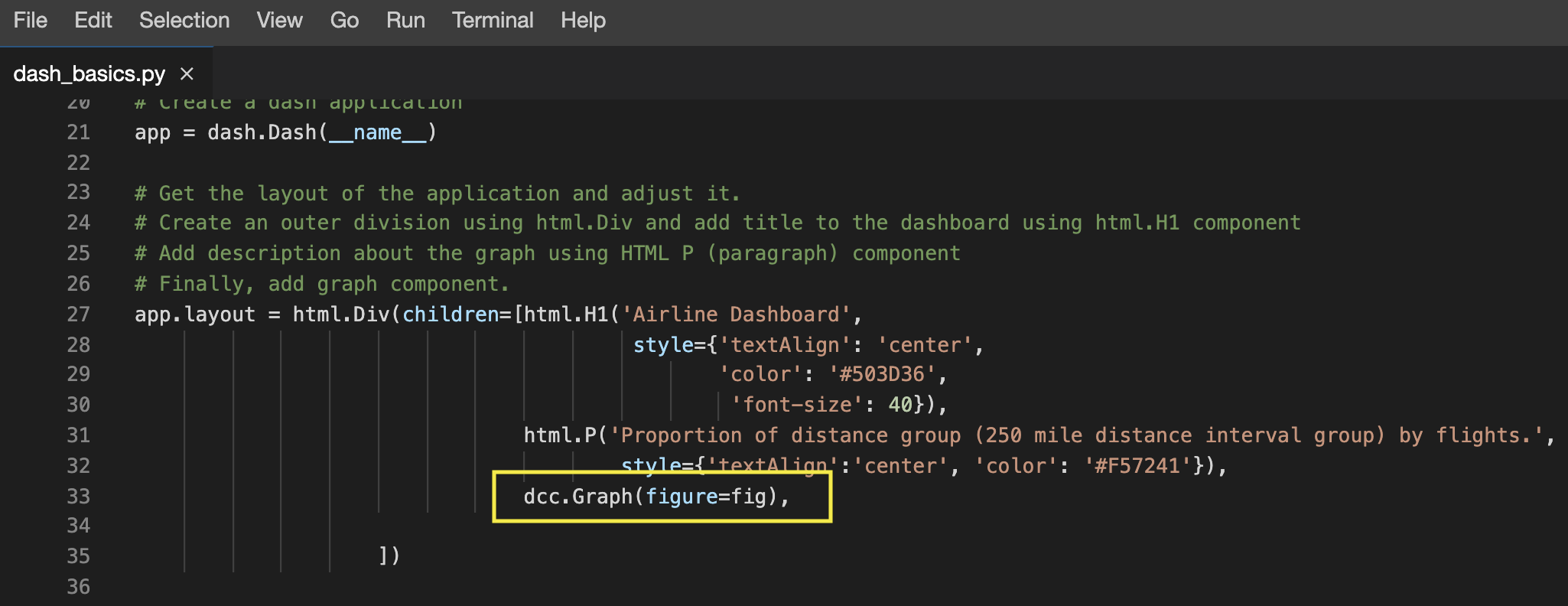


# TASK 5 - Update the graph

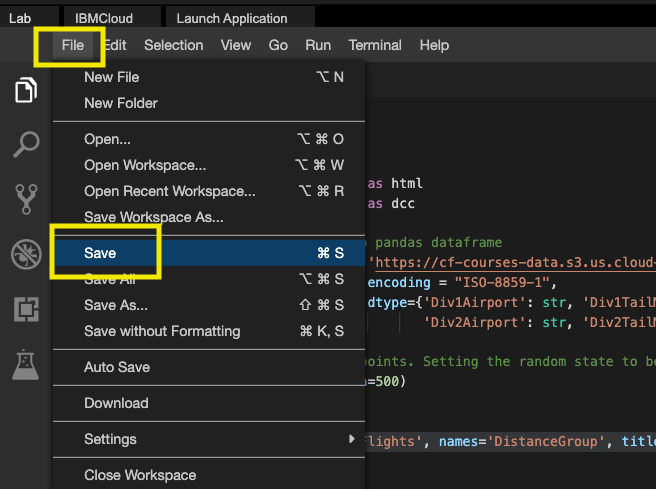
Update figure parameter of dcc.Graph() component to add the pie chart. We have created pie chart and assigned it to fig. Let's use that to update the figure parameter.

**dcc**.Graph(figure=fig)

After updating the dcc.Graph() with the application title, the app.layout will look like:



Before running the application, save the file by clicking on **File -> Save** from the menu bar.



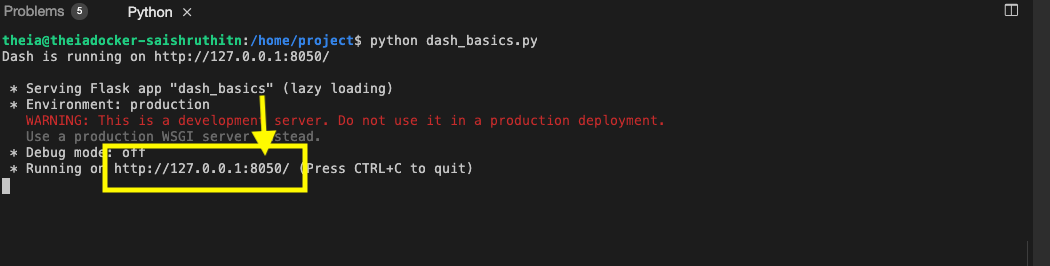
Refer [**here**](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DV0101EN-SkillsNetwork/labs/Module%204/4.5_Dash_Basics.py) to know how your python code should look like.

**TASK 6 - Run the application**

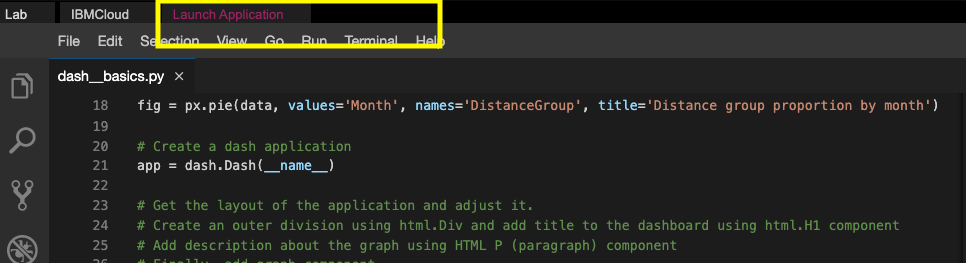
* Run the python file using the following command in the terminal

**python3** **dash\_basics**.py

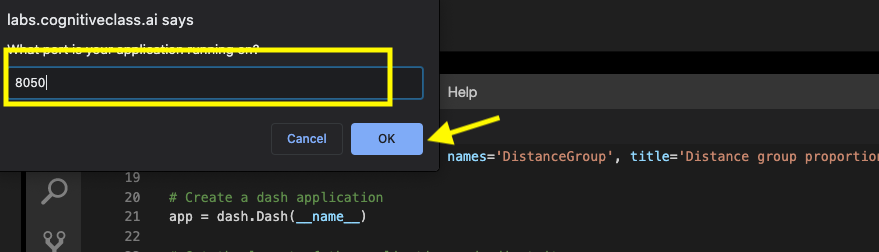
* Observe the port number shown in the terminal.



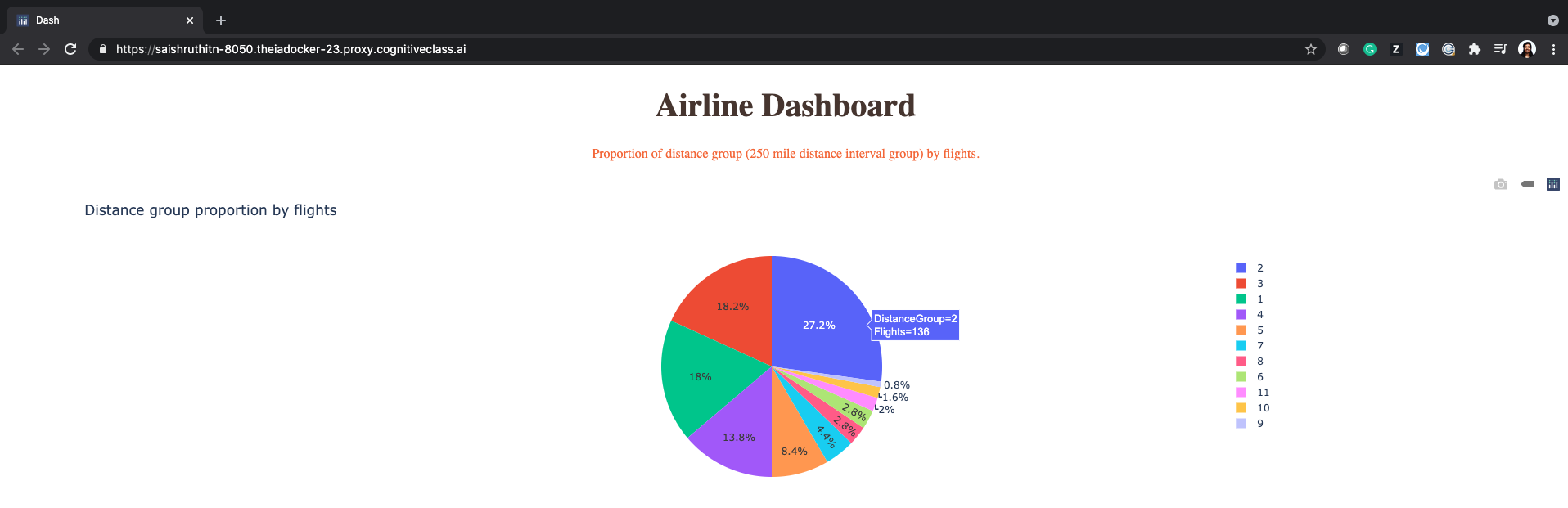
* Click on the Launch Application option from the menu bar.



* Provide the port number and click OK



The app will open in a new browser tab like below:



**Congratulations, you have successfully created your first dash application!**