

Conceptos básicos de Dash: HTML y componentes principales



Objetivos

Después de completar el laboratorio, usted podrá:

- Crear un diseño de aplicación de tablero
- Agregar componentes HTML H1, P y Div
- Agregar componente gráfico principal
- Agregar varios gráficos

Tiempo estimado necesario: 30 minutos

Conjunto de datos utilizado

[Conjunto de datos de informes de puntualidad de aerolíneas](#) de [Data Asset eXchange](#)

Acerca de Skills Network Cloud IDE

Este entorno de desarrollo integrado (IDE) en la nube de Skills Network Labs ofrece un entorno práctico en su navegador web para realizar laboratorios relacionados con cursos y proyectos. Utiliza Theia, una plataforma IDE de código abierto, que puede ejecutarse en computadoras de escritorio o en la nube.

Hasta ahora, en el curso, ha utilizado Jupyter Notebooks para ejecutar su código Python. Este IDE ofrece una alternativa para editar y ejecutar su código Python. En este laboratorio, utilizará este entorno de ejecución alternativo de Python para crear y ejecutar sus aplicaciones Dash.

Aviso importante sobre este entorno de laboratorio

Tenga en cuenta que las sesiones de este entorno de laboratorio no se conservan. Al iniciar el IDE en la nube, se le mostrará un equipo dedicado en la nube exclusivamente para usted. Este equipo estará disponible mientras trabaje activamente en los laboratorios.

Al cerrar su sesión o si se agota por inactividad, se cerrará su sesión y esta "computadora dedicada en la nube" se eliminará junto con cualquier archivo que haya creado, descargado o instalado. La próxima vez que inicie este laboratorio, se creará un nuevo entorno.

Si solo completas una parte del laboratorio y regresas más tarde, podrías tener que empezar desde el principio. Por lo tanto, es recomendable planificar tu tiempo adecuadamente y completar los laboratorios en una sola sesión.

Comencemos a crear la aplicación Dash

Meta

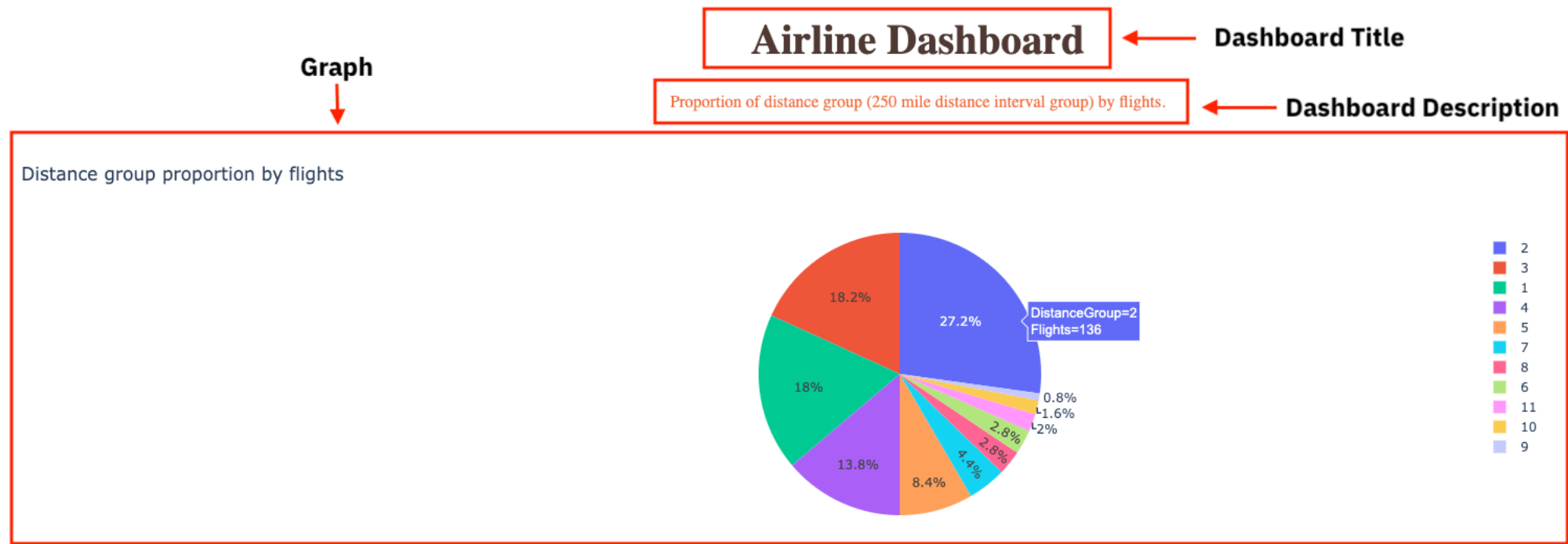
Cree un panel que muestre el porcentaje de vuelos que se ejecutan en un grupo de distancia específico. El grupo de distancia corresponde a los intervalos de distancia, cada 250 millas, para cada segmento de vuelo. Si el vuelo cubre hasta 500 millas, se incluirá en el grupo de distancia 2 (250 millas + 250 millas).

Resultado esperado

A continuación se muestra el resultado esperado del laboratorio. Nuestra aplicación de panel de control consta de tres componentes:

- Título de la solicitud
- Descripción de la aplicación

- Gráfico que muestra la proporción del grupo de distancia por mes



Hacer:

1. Importar las bibliotecas necesarias y leer el conjunto de datos
2. Crear un diseño de aplicación
3. Agregar título al panel de control mediante el componente HTML H1
4. Agregar un párrafo sobre el gráfico usando el componente HTML P
5. Agregar el gráfico circular anterior usando el componente gráfico principal
6. Ejecutar la aplicación

Prepare la herramienta

- Instala los paquetes de Python necesarios para ejecutar la aplicación. Copia y pega el siguiente comando en la terminal.

```
python3.8 -m pip install packaging
```

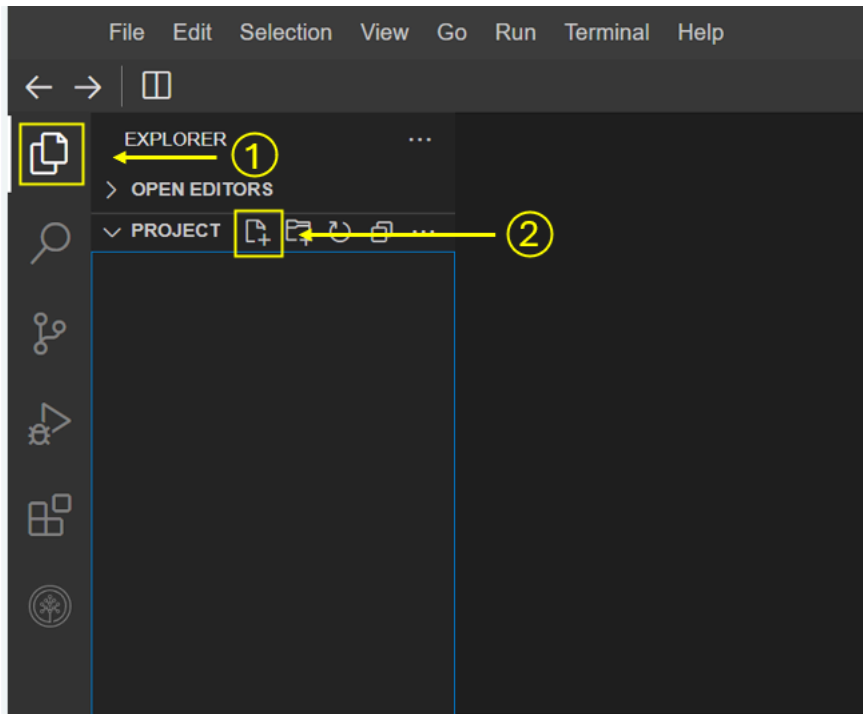
```
python3.8 -m pip install pandas dash
```

```
theia@theiadocker-malikas: /home/project x
theia@theiadocker-malikas: /home/project$ python3 -m pip install pandas dash
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/c3/e2/00cacecafbab071c787019f00ad84ca3185952f6bb9bca9550ed83870d4d/pandas-1.1.5-cp36-cp36m-manylinux1_x86_64.whl (9.5MB)
    100% |#####| 9.5MB 163kB/s
Collecting dash
  Downloading https://files.pythonhosted.org/packages/cc/42/e1692b2d34e4135569db680efe3438e809a6b3f0ae607ad41aeff7741672/dash-2.6.1-py3-none-any.whl (9.9MB)
    100% |#####| 9.9MB 159kB/s
Collecting pytz>=2017.2 (from pandas)
  Cache entry deserialization failed, entry ignored
  Downloading https://files.pythonhosted.org/packages/d5/50/54451e88e3da4616286029a3a17fc377de817f66a0f50e1faaee90161724/pytz-2022.2.1-py2.py3-none-any.whl (501kB)
    100% |#####| 501kB 3.2MB/s
Collecting python-dateutil>=2.7.3 (from pandas)
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading https://files.pythonhosted.org/packages/36/7a/87837f39d0296e723bb9b62bbb257d0355c7f6128853c78955f57342a56d/python_dateutil-2.8.2-py2.py3-none-any.whl (247kB)
    100% |#####| 256kB 5.8MB/s
Collecting numpy>=1.15.4 (from pandas)
  Downloading https://files.pythonhosted.org/packages/45/b2/6c7545bb7a38754d63048c7696804a0d947328125d81bf12beaa692c3ae3/numpy-1.19.5-cp36-cp36m-manylinux1_x86_64.whl (13.4MB)
    100% |#####| 13.4MB 111kB/s
Collecting contextvars==2.4; python_version < "3.7" (from dash)
  Downloading https://files.pythonhosted.org/packages/83/96/55b82d9f13763be9d672622e1b8106c85acb83edd7cc2fa5bc67cd9877e9/contextvars-2.4.tar.gz
Collecting dash-table==5.0.0 (from dash)
  Downloading https://files.pythonhosted.org/packages/da/ce/43f77dc8e7bbad02a9f88d07bf794eaf68359df756a28bb9f2f78e255bb1/dash_table-5.0.0-py3-none-any.whl
```

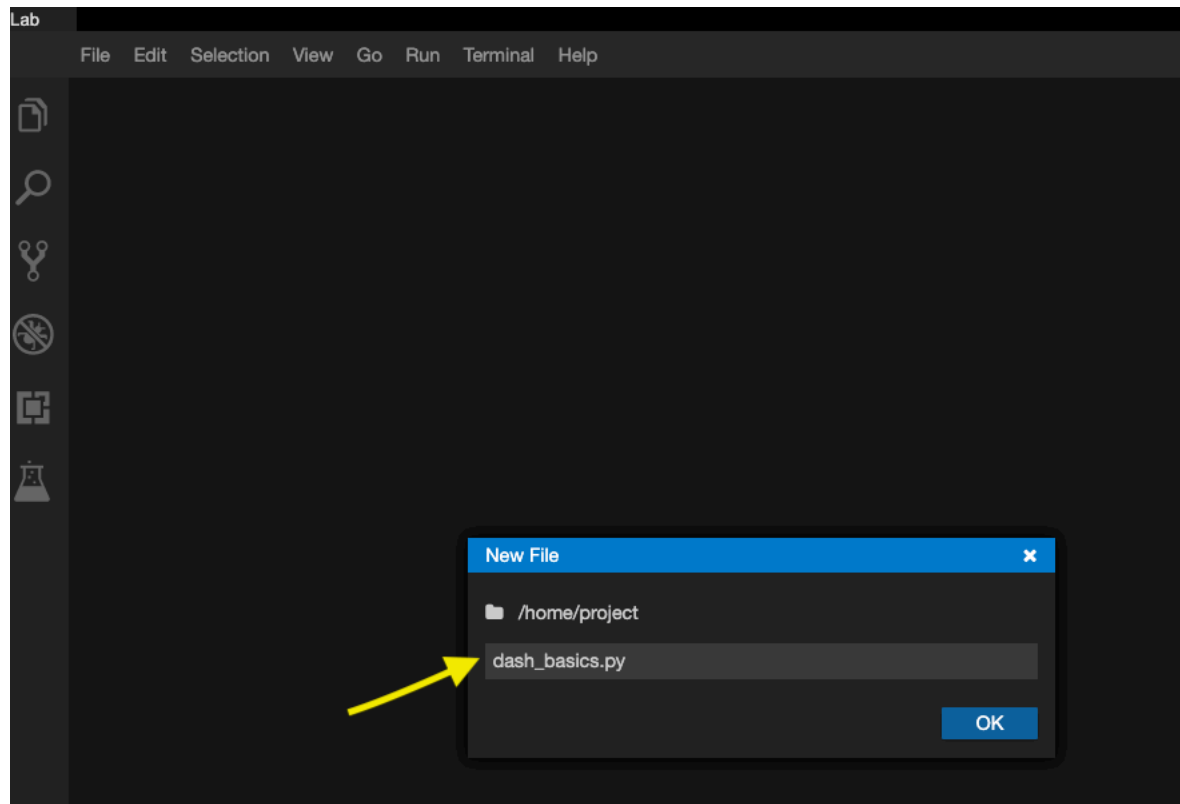
```
pip3 install httpx==0.20 dash plotly
```

```
theia@theiadocker-malikas: /home/project x
theia@theiadocker-malikas: /home/projects$ pip3 install httpx==0.20 dash plotly
/usr/lib/python3/dist-packages/secretstorage/decrypt.py:15: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/usr/lib/python3/dist-packages/secretstorage/util.py:19: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Defaulting to user installation because normal site-packages is not writeable
Collecting httpx==0.20
  Downloading httpx-0.20.0-py3-none-any.whl (82 kB)
    |#####| 82 kB 779 kB/s
Collecting dash
  Downloading dash-2.6.1-py3-none-any.whl (9.9 MB)
    |#####| 9.9 MB 40.7 MB/s
Collecting plotly
  Downloading plotly-5.10.0-py2.py3-none-any.whl (15.2 MB)
    |#####| 15.2 MB 39.3 MB/s
Requirement already satisfied: sniffio in /home/theia/.local/lib/python3.6/site-packages (from httpx==0.20) (1.2.0)
Requirement already satisfied: httpcore<0.14.0,>=0.13.3 in /home/theia/.local/lib/python3.6/site-packages (from httpx==0.20) (0.13.7)
Requirement already satisfied: async-generator in /home/theia/.local/lib/python3.6/site-packages (from httpx==0.20) (1.10)
Requirement already satisfied: certifi in /home/theia/.local/lib/python3.6/site-packages (from httpx==0.20) (2020.12.5)
Requirement already satisfied: rfc3986[idna2008]<2,>=1.3 in /home/theia/.local/lib/python3.6/site-packages (from httpx==0.20) (1.5.0)
Requirement already satisfied: charset-normalizer in /home/theia/.local/lib/python3.6/site-packages (from httpx==0.20) (2.0.12)
Collecting dash-html-components==2.0.0
  Downloading dash_html_components-2.0.0-py3-none-any.whl (4.1 kB)
Collecting dash-table==5.0.0
  Downloading dash_table-5.0.0-py3-none-any.whl (3.9 kB)
```

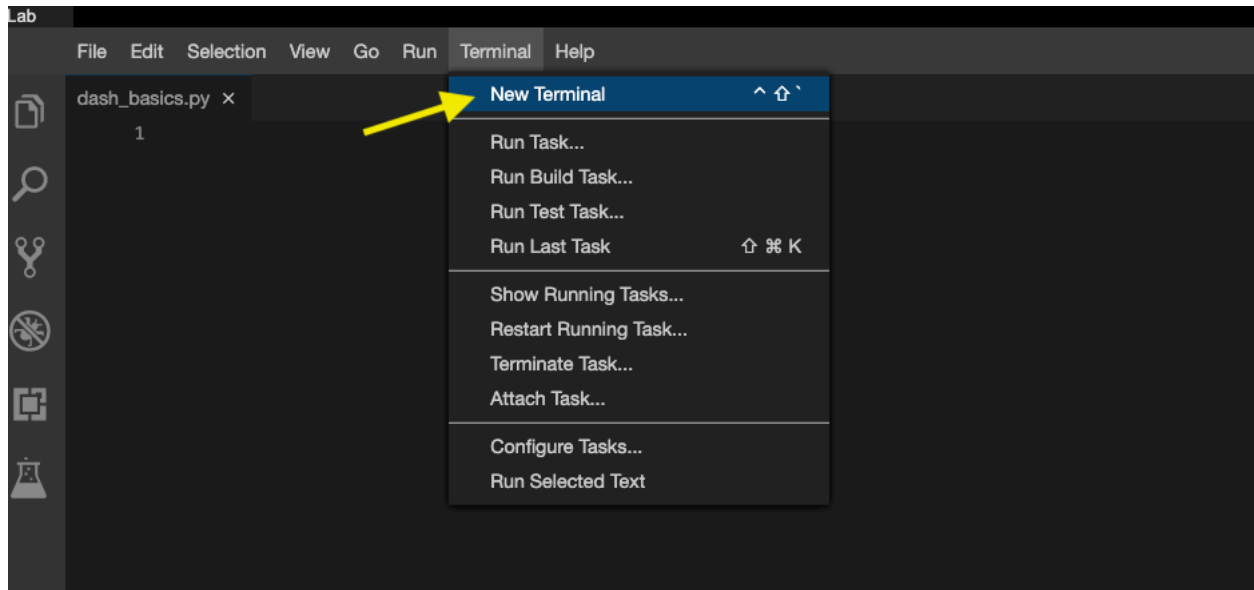
- Cree un nuevo script de Python haciendo clic en el ícono **del explorador** de la barra de herramientas lateral y seleccionando el ícono **de nuevo archivo** , como se muestra en la imagen a continuación.



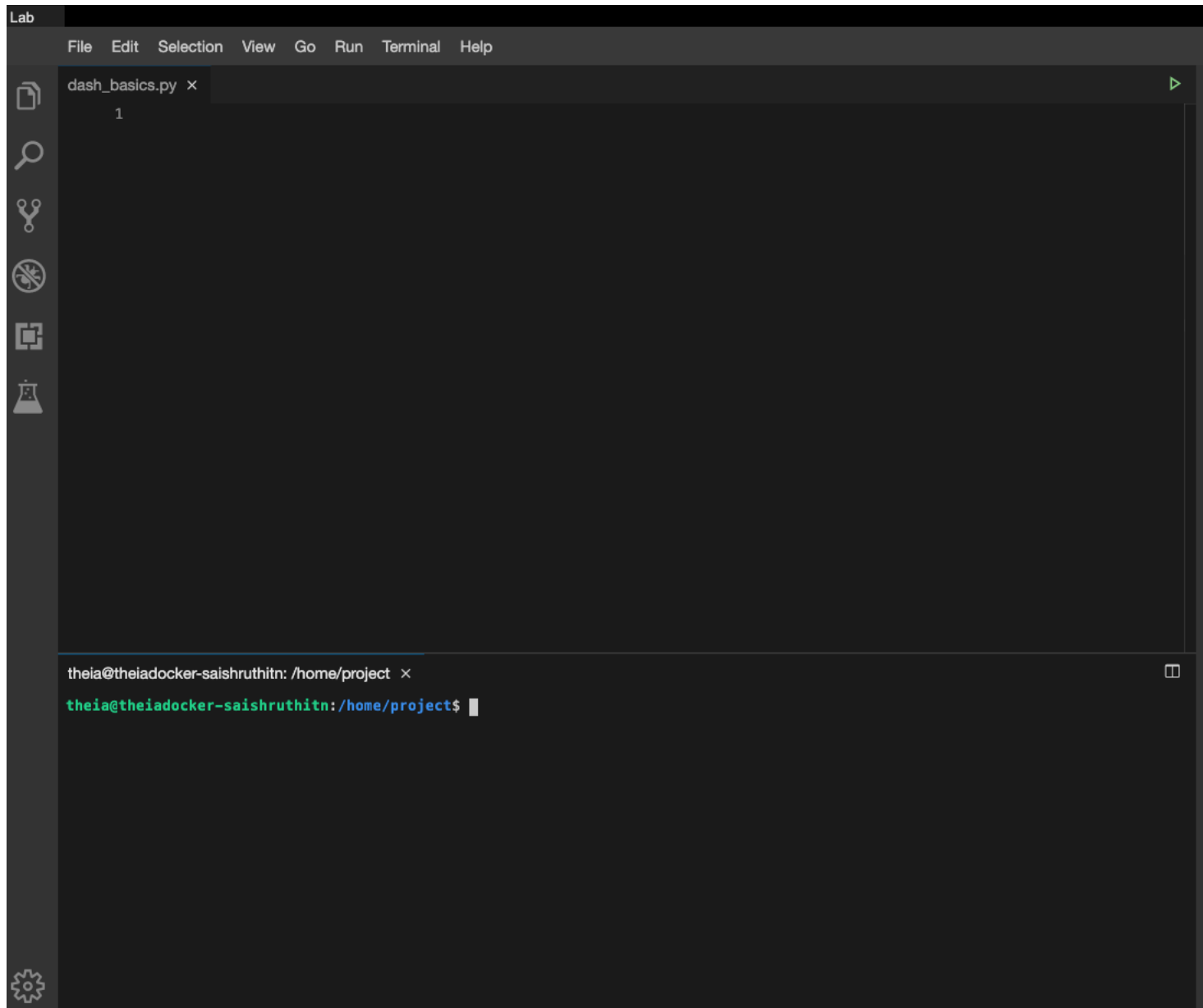
- Proporcione el nombre del archivo comodash_basics.py



- Abra una nueva terminal, haciendo clic en la barra de menú y seleccionando **Terminal** -> **Nueva terminal** , como en la imagen de abajo.



- Ahora ya tienes el script y la terminal listos para iniciar el laboratorio.



TAREA 1 - Preparación de datos

Vamos a empezar con

- Importación de las bibliotecas necesarias
- Lectura y muestreo de 500 puntos de datos aleatorios

- Prepare el gráfico

Copie el código a continuación en el `dash_basics.pyscript` y revise el código.

```
# Import required packages
import pandas as pd
import plotly.express as px
import dash
from dash import dcc
from dash import html
# 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')
```

TAREA 2: Crear una aplicación de tablero y obtener el esqueleto del diseño

A continuación, creamos el esqueleto de nuestra aplicación de tablero. Nuestra aplicación de tablero tiene tres componentes, como se vio anteriormente:

- Título de la solicitud
- Descripción de la aplicación
- Gráfico que muestra la proporción del grupo de distancia por mes

Asignación a las respectivas etiquetas HTML de Dash:

- Título añadido mediante `html.H1()` etiqueta
- Descripción añadida mediante `html.P()` etiqueta
- Gráfico añadido mediante `dcc.Graph()` etiqueta

Copie el siguiente código en el `dash_basics.pyscript` y revise la estructura.

NOTA : Copie a continuación el código actual

```
# 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()
```

TAREA 3 - Agregar el título de la aplicación

Actualice la `html.H1()` etiqueta para contener el título de la aplicación.

- El título de la aplicación es `Airline Dashboard`
- Utilice el parámetro de estilo proporcionado a continuación para centralizar el título, con el código de color `#503D36` y el tamaño de fuente como se indica. `40`

```
'Airline Dashboard',style={'text-align': 'center', 'color': '#503D36', 'font-size': 40}
```

Después de actualizar `html.H1()` con el título de la aplicación, `app.layout` se verá así:

dash_basics.py ×

```

20 # Create a dash application
21 app = dash.Dash(__name__)
22
23 # Get the layout of the application and adjust it.
24 # Create an outer division using html.Div and add title to the dashboard using html.H1
25 # Add description about the graph using HTML P (paragraph) component
26 # Finally, add graph component
27 app.layout = html.Div(children=[html.H1('Airline Dashboard',
28                                     style={'textAlign': 'center',
29                                     'color': '#503D36',
30                                     'font-size': 40}),
31                                html.P(),
32                                dcc.Graph(),
33                                ],)
34
35

```

TAREA 4 - Agregar la descripción de la aplicación

Actualice la `html.P()` etiqueta para contener la descripción de la aplicación.

- La descripción es `Proportion of distance group (250 mile distance interval group) by flights.`
- Utilice el parámetro de estilo para hacer que la descripción center esté alineada y con color `#F57241`.

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

Después de actualizar `html.H1()` con el título de la aplicación, `app.layout` se verá así:

```
File Edit Selection View Go Run Terminal Help

dash_basics.py ●
20 # Create a dash application
21 app = dash.Dash(__name__)
22
23 # Get the layout of the application and adjust it.
24 # Create an outer division using html.Div and add title to the dashboard using html.H1
25 # Add description about the graph using HTML P (paragraph) component
26 # Finally, add graph component.
27 app.layout = html.Div(children=[html.H1('Airline Dashboard',
28                                         style={'textAlign': 'center',
29                                               'color': '#503D36',
30                                               'font-size': 40}),
31                                html.P('Proportion of distance group (250 mile distance
32                                       style={'textAlign': 'center', 'color': '#F57241'})
33                                dcc.Graph(),
34                                ],
35                                style={'background-color': '#f0f0f0'})
36
```

TAREA 5 - Actualizar el gráfico

Actualizar `figure` el parámetro del `dcc.Graph()` componente para agregar el gráfico circular. Hemos creado un gráfico circular y lo hemos asignado a `fig`. Utilicémoslo para actualizar el `figure` parámetro.

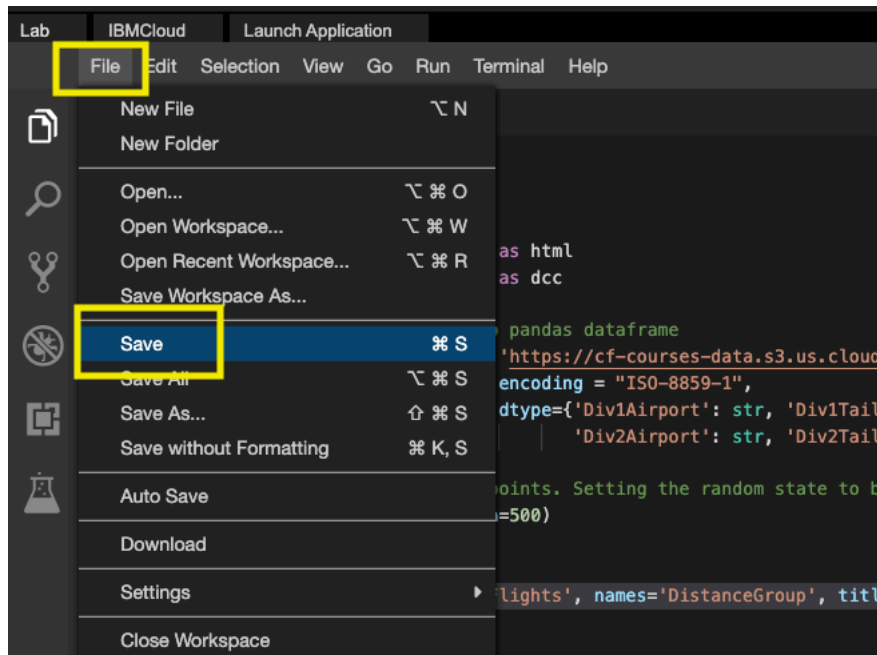
```
figure=fig
```

Después de actualizar `dcc.Graph()` con el título de la aplicación, `app.layout` verá así:

```
File Edit Selection View Go Run Terminal Help

dash_basics.py ×
20 # Create a dash application
21 app = dash.Dash(__name__)
22
23 # Get the layout of the application and adjust it.
24 # Create an outer division using html.Div and add title to the dashboard using html.H1
25 # Add description about the graph using HTML P (paragraph) component
26 # Finally, add graph component.
27 app.layout = html.Div(children=[html.H1('Airline Dashboard',
28                                     style={'textAlign': 'center',
29                                           'color': '#503D36',
30                                           'font-size': 40}),
31                                html.P('Proportion of distance group (250 mile distance
32                                     style={'textAlign': 'center', 'color': '#F57241'
33                                dcc.Graph(figure=fig),
34
35                                ]),
36
```

Antes de ejecutar la aplicación, guarde el archivo haciendo clic en **Archivo** -> **Guardar** en la barra de menú.



Puedes consultar el código completo de Python aquí

```
# Import required packages
import pandas as pd
import plotly.express as px
import dash
from dash import dcc
from dash import html

# 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')

# 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('Airline Dashboard', style={'textAlign': 'center', 'color': '#503D36', 'font-size': 40}),
                                html.P('Proportion of distance group (250 mile distance interval group) by flights.', style={'textAlign': 'center', 'color': '#F57241'}),
                                dcc.Graph(figure=fig),
                                ])

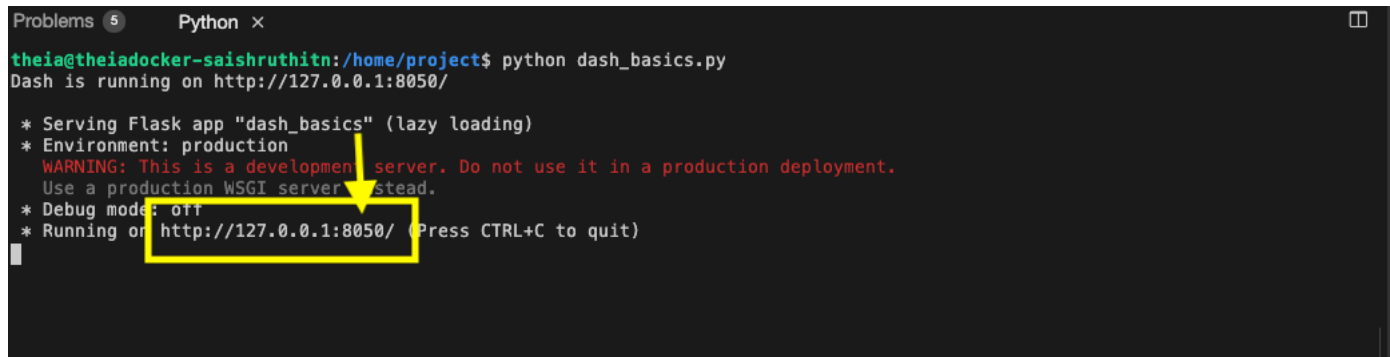
# Run the application
if __name__ == '__main__':
    app.run()
```

TAREA 6 - Ejecutar la aplicación

- Ejecute el archivo de Python usando el siguiente comando en la terminal

```
python3.8 dash_basics.py
```

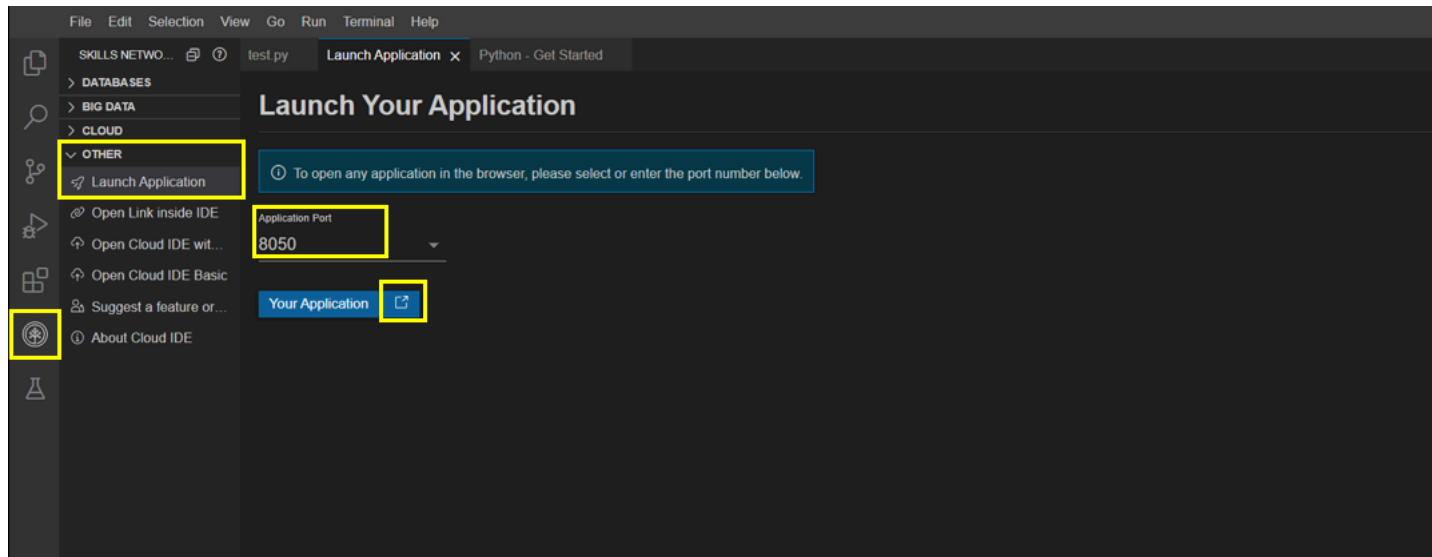
- Observe the port number shown in the terminal.



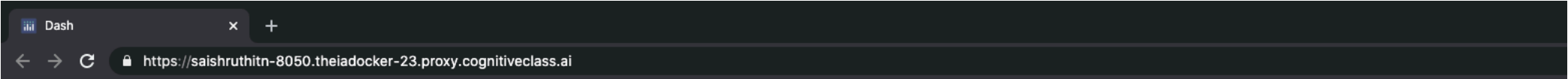
```
Problems 5 Python x
theia@theiadocker-saishruthitn:/home/project$ python dash_basics.py
Dash is running on http://127.0.0.1:8050/

* Serving Flask app "dash_basics" (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: off
* Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
```

- Click on the Launch Application option from the side menu bar. Provide the port number and click OK



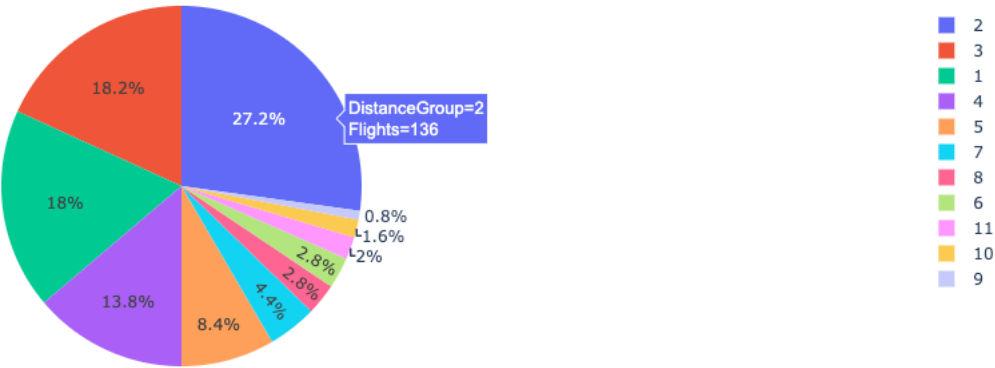
The app will open in a new browser tab like below:



Airline Dashboard

Proportion of distance group (250 mile distance interval group) by flights.

Distance group proportion by flights



Congratulations, you have successfully created your first dash application!

Exercise : Practice Tasks

You will practice some tasks to update the dashboard.

- Change the title to the dashboard from “Airline Dashboard” to “Airline On-time Performance Dashboard” using HTML H1 component and font-size as 50.
► Answer
- Save the above changes and relaunch the dashboard application to see the updated dashboard title.
► Answer
- Write a command to stop the running app in the terminal
► Answer

Author

[Saishruthi Swaminathan](#)

