Dash Bileşenleri

hedefler

Laboratuvarı tamamladıktan sonra şunları yapabileceksiniz:

- Kısa çizgi uygulama düzeni oluşturun
- HTML H1, P ve Div bileşenlerini ekleyin
- Çekirdek grafik bileşeni ekle
- Birden fazla grafik ekle

Gerekli tahmini süre: 30 dakika

Kullanılan Veri Kümesi

Data Asset eXchange'ten Havayolu Raporlama Taşıyıcı Zamanında Performans veri kümesi

Skills Network Cloud IDE Hakkında

Bu Skills Network Labs Cloud IDE (Entegre Geliştirme Ortamı), kurs ve proje ile ilgili laboratuvarları tamamlamak için web tarayıcınızda uygulamalı bir ortam sağlar. Masaüstünde veya bulutta çalıştırılabilen açık kaynaklı bir IDE platformu olan Theia'yı kullanır. Kursta şu ana kadar python kodunuzu çalıştırmak için Jupyter not defterlerini kullandınız. Bu IDE, Python kodunuzu düzenlemek ve çalıştırmak için bir alternatif sunar. Bu laboratuvarda, Dash uygulamalarınızı oluşturmak ve başlatmak için bu alternatif Python çalışma zamanını kullanacaksınız.

Bu laboratuvar ortamı hakkında Önemli Bildirim

Lütfen bu laboratuvar ortamı için oturumların kalıcı olmadığını unutmayın. Cloud IDE'yi başlattığınızda, size özel olarak 'bulut üzerinde özel bir bilgisayar' sunulur. Bu, laboratuvarlarda aktif olarak çalıştığınız sürece sizin için kullanılabilir.

Oturumunuzu kapattığınızda veya etkinlik olmaması nedeniyle zaman aşımına uğradığında, oturumunuz kapatılır ve bu 'buluttaki özel bilgisayar', oluşturmuş, indirmiş veya kurmuş olabileceğiniz tüm dosyalarla birlikte silinir. Bu laboratuvarı bir sonraki başlatışınızda, sizin için yeni bir ortam oluşturulur.

Laboratuvarın sadece bir kısmını bitirir ve daha sonra geri dönerseniz, baştan başlamak zorunda kalabilirsiniz. Bu nedenle, zamanınızı buna göre planlamanız ve laboratuvarlarınızı tek seansta bitirmeniz iyi bir fikirdir.

Dash uygulaması oluşturmaya başlayalım

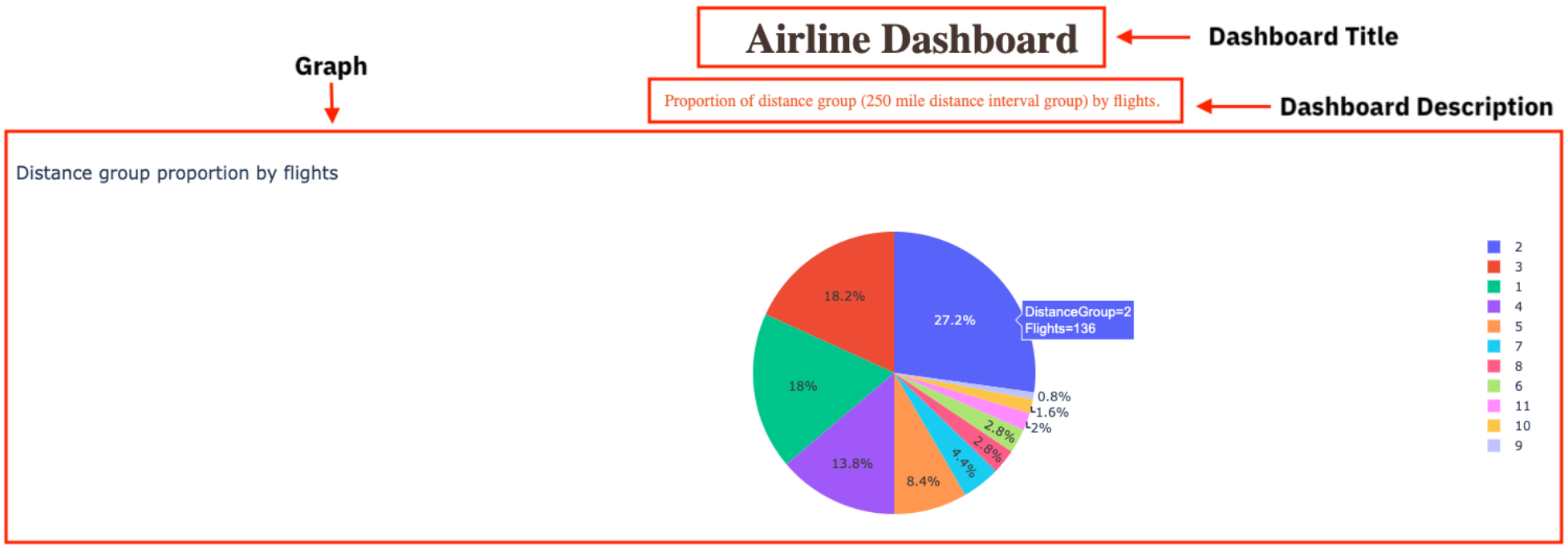
Hedef

Belirli bir mesafe grubu altında yürütülen uçuşların yüzdesini görüntüleyen bir pano oluşturun. Mesafe grubu, uçuş segmenti için her 250 milde bir olan mesafe aralıklarıdır. Uçuş 500 mile gidiyorsa, mesafe grubu 2 (250 mil + 250 mil) altında olacaktır.

Beklenen çıktı

Laboratuvardan beklenen sonuç aşağıdadır. Pano uygulamamız üç bileşenden oluşur:

- Uygulamanın başlığı
- Uygulamanın açıklaması
- Aya göre uzaklık grubunun oranını gösteren grafik



Yapmak:

1. Gerekli kitaplıkları içe aktarın ve veri kümesini okuyun
2. Bir uygulama düzeni oluşturun
3. HTML H1 bileşenini kullanarak panoya başlık ekleyin
4. HTML P bileşenini kullanarak grafik hakkında bir paragraf ekleyin
5. Çekirdek grafik bileşenini kullanarak yukarıdaki pasta grafiği ekleyin
6. Uygulamayı çalıştırın

Aracı hazırlayın

- Uygulamayı çalıştırmak için gerekli python paketlerini kurun. Aşağıdaki komutu kopyalayıp terminale yapıştırın.

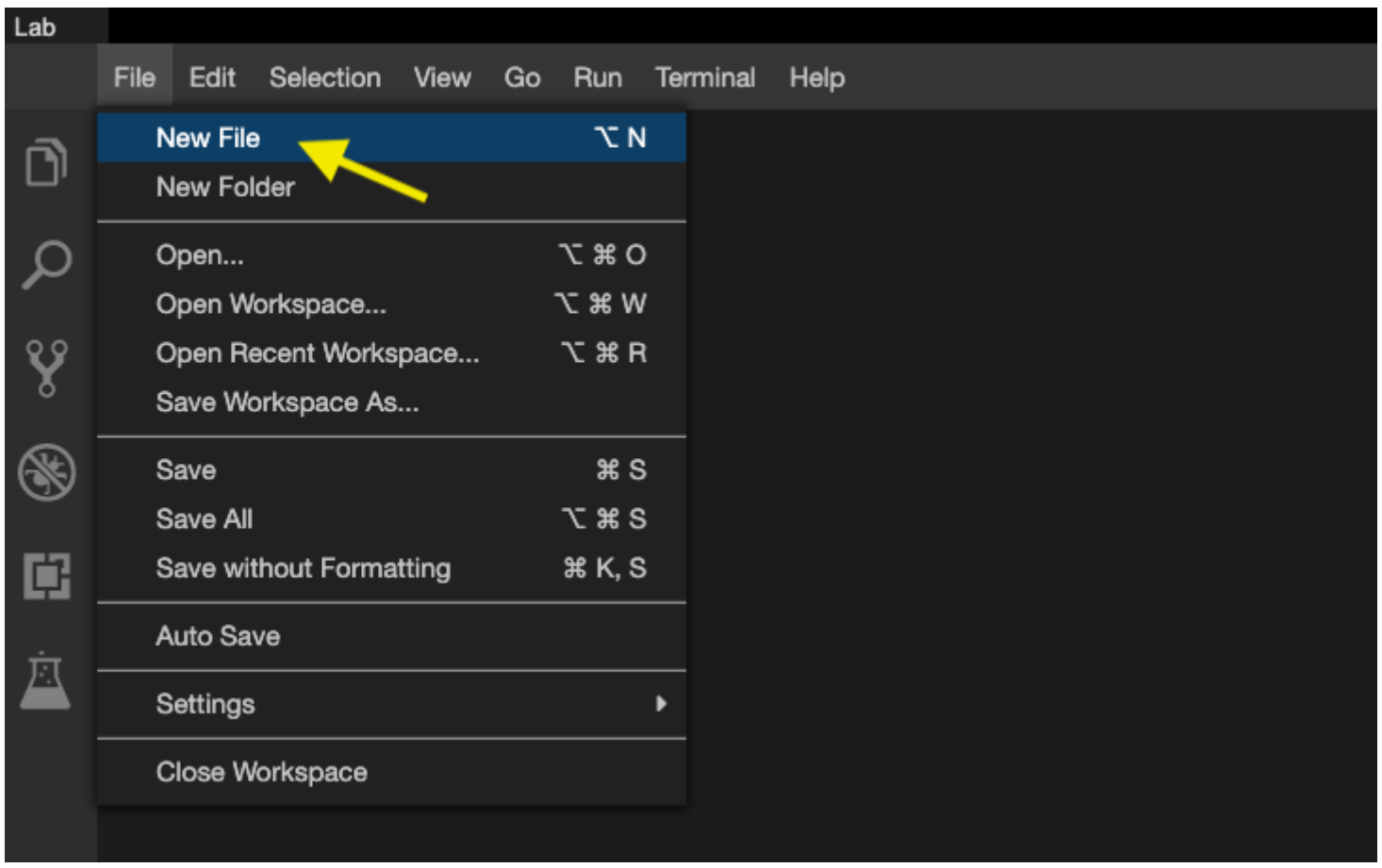
```
python3 -m pip install pandas dash
```

```
theia@theiadocker-malikas: /home/project
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_x8
hl (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 (5
    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-an
(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_x8
hl (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
```

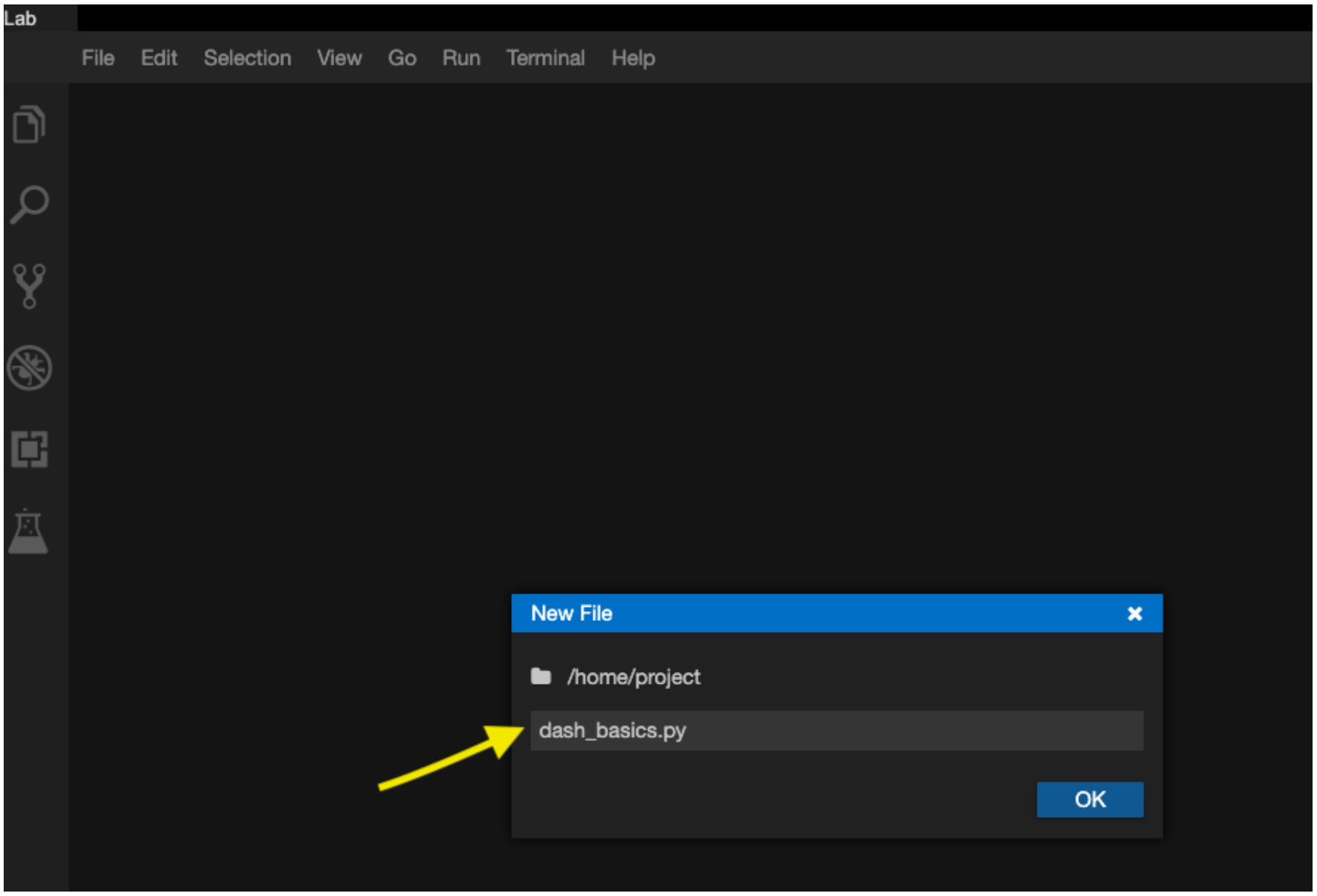
```
pip3 install httpx==0.20 dash plotly
```

```
theia@theiadocker-malikas: /home/project ×

theia@theiadocker-malikas:/home/project$ pip3 install httpx==0.20 dash plotly
/usr/lib/python3/dist-packages/secretstorage/dhcrypto.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_byt
es 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-pack
ages (from httpx==0.20) (0.13.7)
Requirement already satisfied: async-generator in /home/theia/.local/lib/python3.6/site-packages (fro
m 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-pac
kages (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)
```

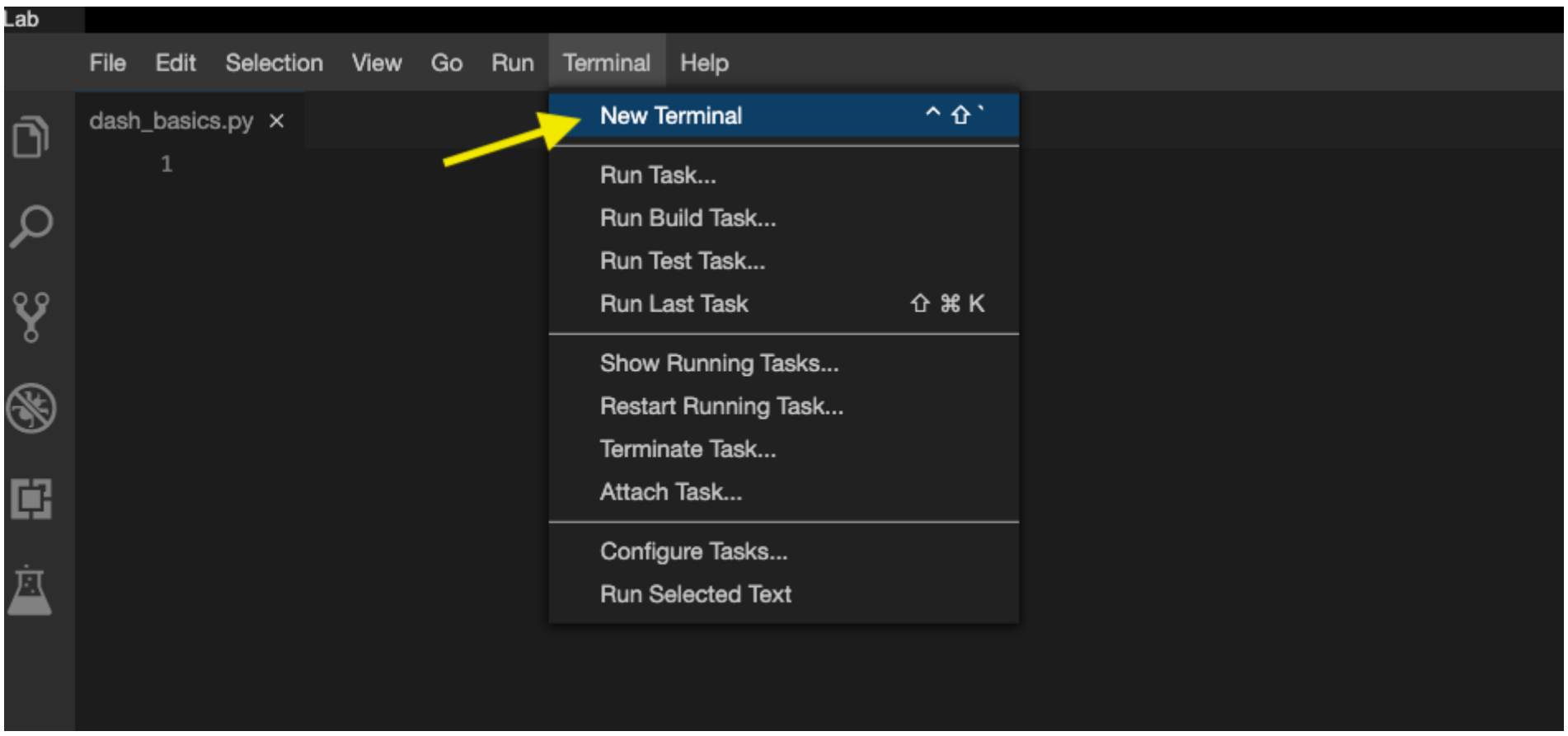
- Aşağıdaki görüntüdeki gibi menü çubuğuna tıklayıp **File** -> **New File seçerek yeni bir python betiği oluşturun.**



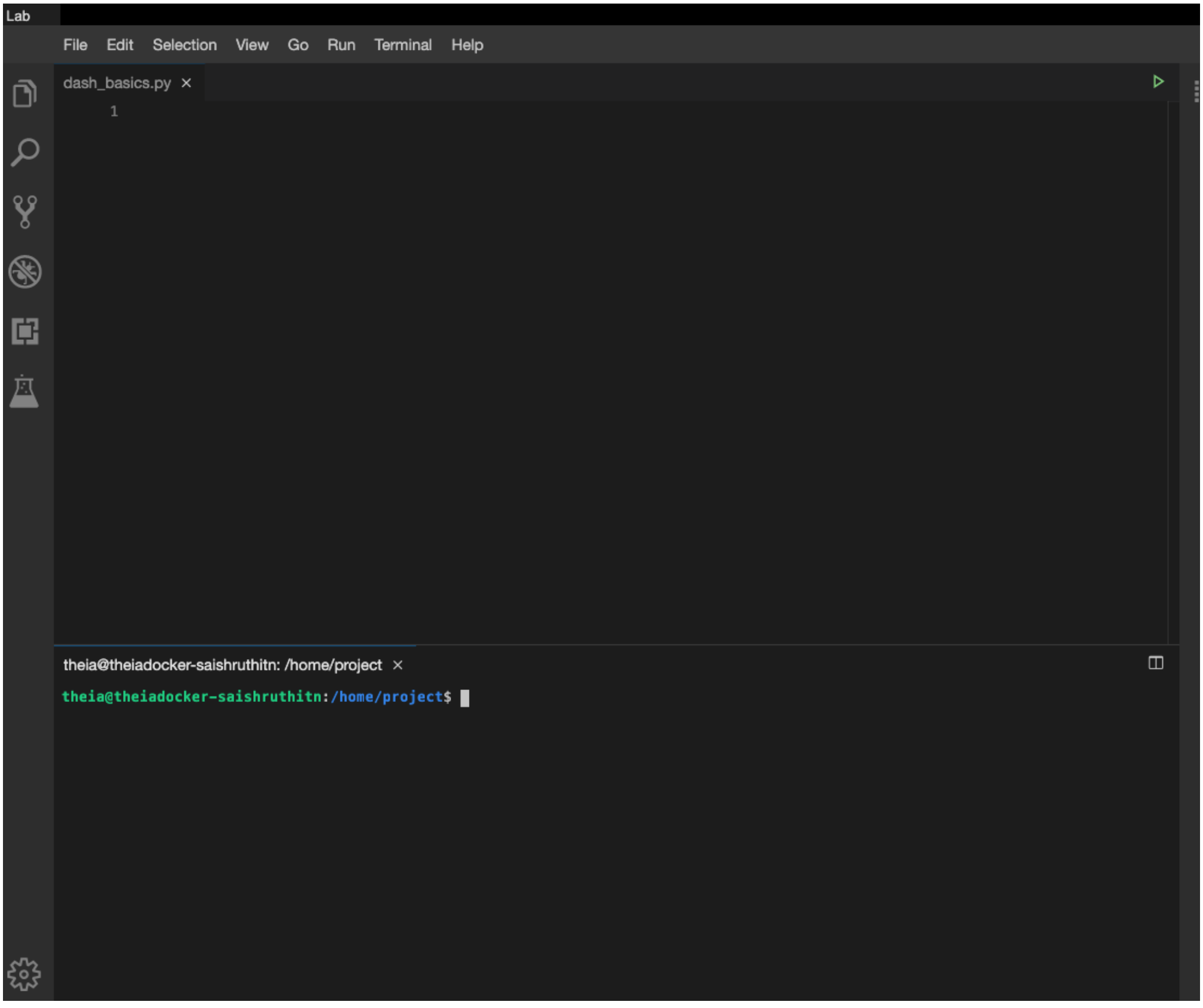
- Dosya adını şu şekilde sağlayın:`dash_basics.py`



- Aşağıdaki görüntüdeki gibi menü çubuğuna tıklayıp **Terminal** -> **New Terminal öğesini seçerek yeni bir terminal açın.**



- Artık, laboratuvarı başlatmak için komut dosyanız ve terminaliniz hazır.



GÖREV 1 - Veri Hazırlama

İle başlayalım

- Gerekli kitaplıkları içe aktarma
- 500 rastgele veri noktasını okuma ve örnekleme
- Tabloyu hazırlayın

Aşağıdaki kodu `dash_basics.py`betiğe kopyalayın ve kodu inceleyin.

```
# 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')
```

GÖREV 2 - Kısa çizgi uygulaması oluşturun ve düzen iskeletini alın

Ardından, tire uygulamamız için bir iskelet oluşturuyoruz. Pano uygulamamızın daha önce görüldüğü gibi üç bileşeni vardır:

- Uygulamanın başlığı
- Uygulamanın açıklaması
- Aya göre uzaklık grubunun oranını gösteren grafik

İlgili Dash HTML etiketleriyle eşleme:

- `html.H1()`Etiket kullanılarak eklenen başlık
- Açıklama `html.P()`etiket kullanılarak eklendi
- 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:

dash_basics.py

```
# 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(),
                                dcc.Graph(),

                                ])

```

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`.

```
('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:

File Edit Selection View Go Run Terminal Help

dash_basics.py

```
# 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(),

                                ])

```

TASK 5 - Update the graph

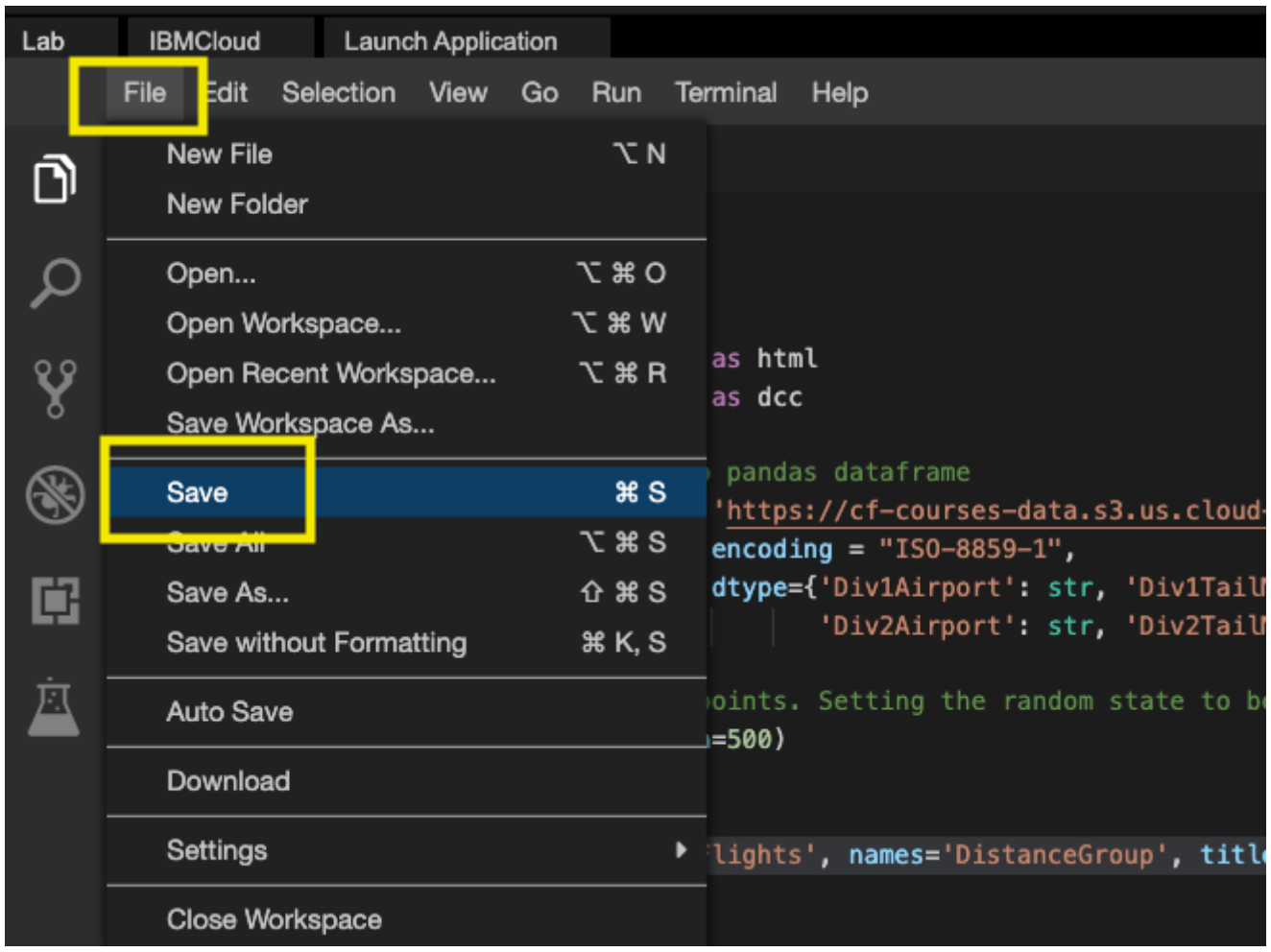
Update `figure` parameter of `dcc.Graph()` component to add the pie chart. We have created pie chart and assigned it to `fig`. Let's use that to update the `figure` parameter.

```
(figure=fig)
```

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

```
File  Edit  Selection  View  Go  Run  Terminal  Help

dash_basics.py ×

    # 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),

                                    ])

```

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



You can Refer to the entire python code here

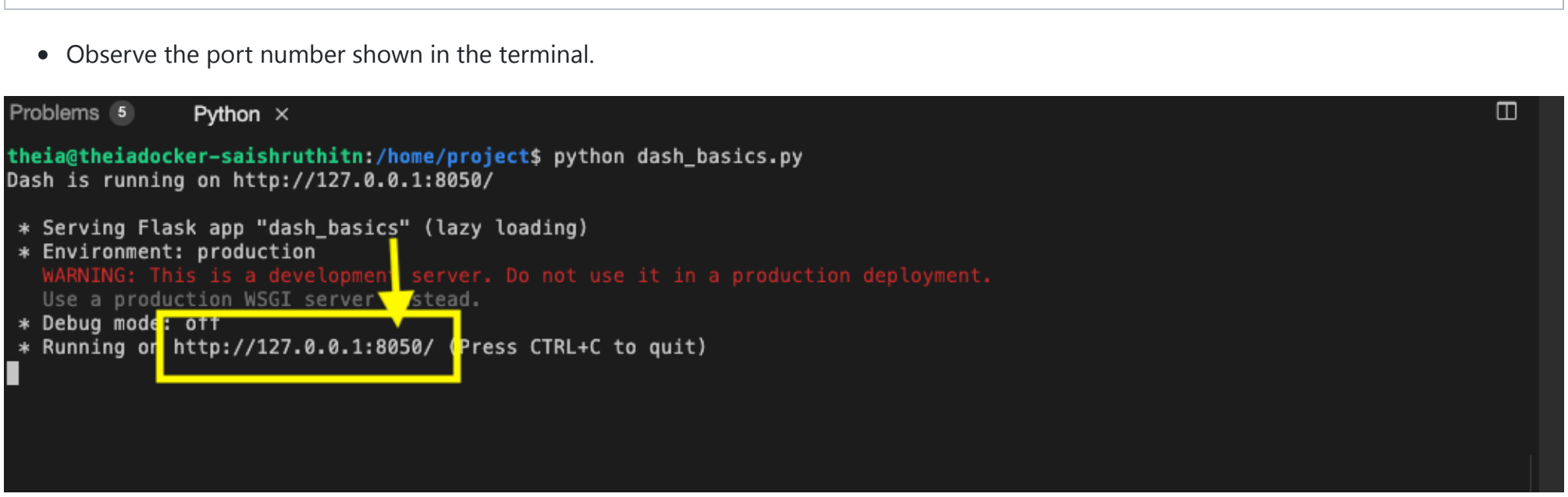
```
# Import required packages
import pandas as pd
import plotly.express as px
import dash
import dash_html_components as html
import dash_core_components as dcc

# 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')

# 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_server()
```

TASK 6 - Run the application

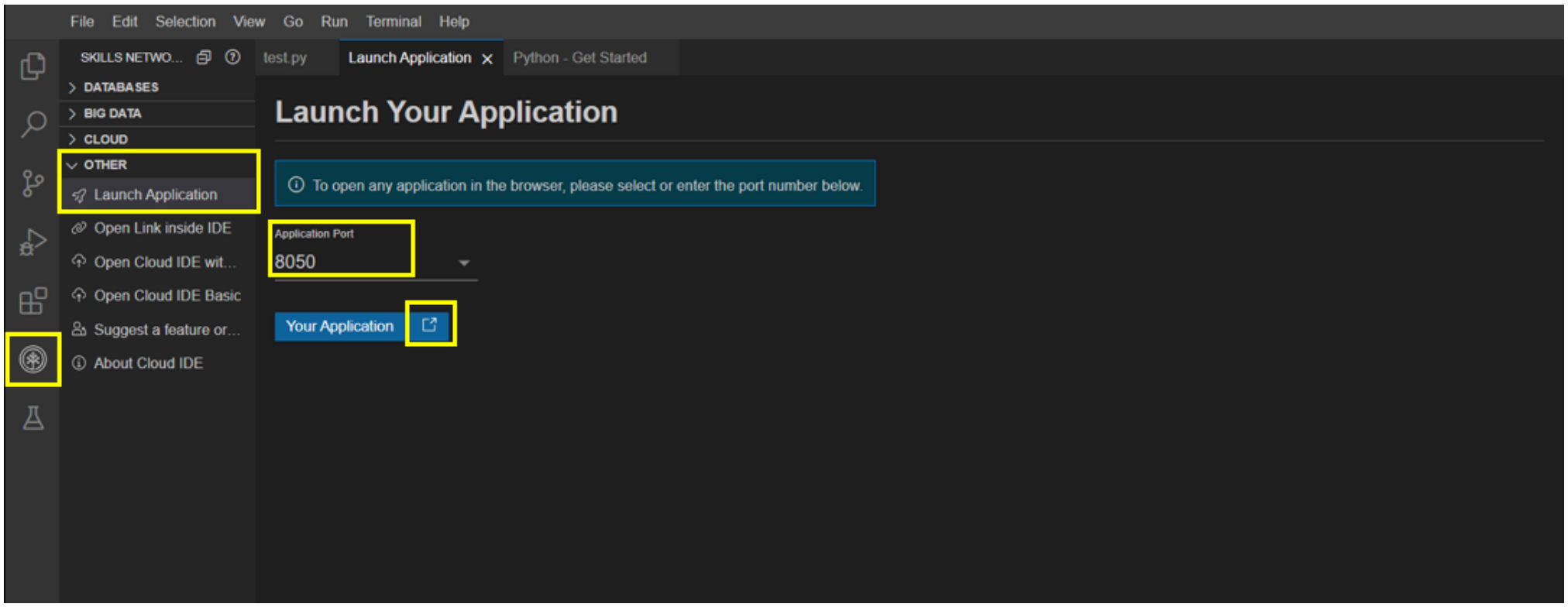
- Run the python file using the following command in the terminal

```
python3 dash_basics.py
```

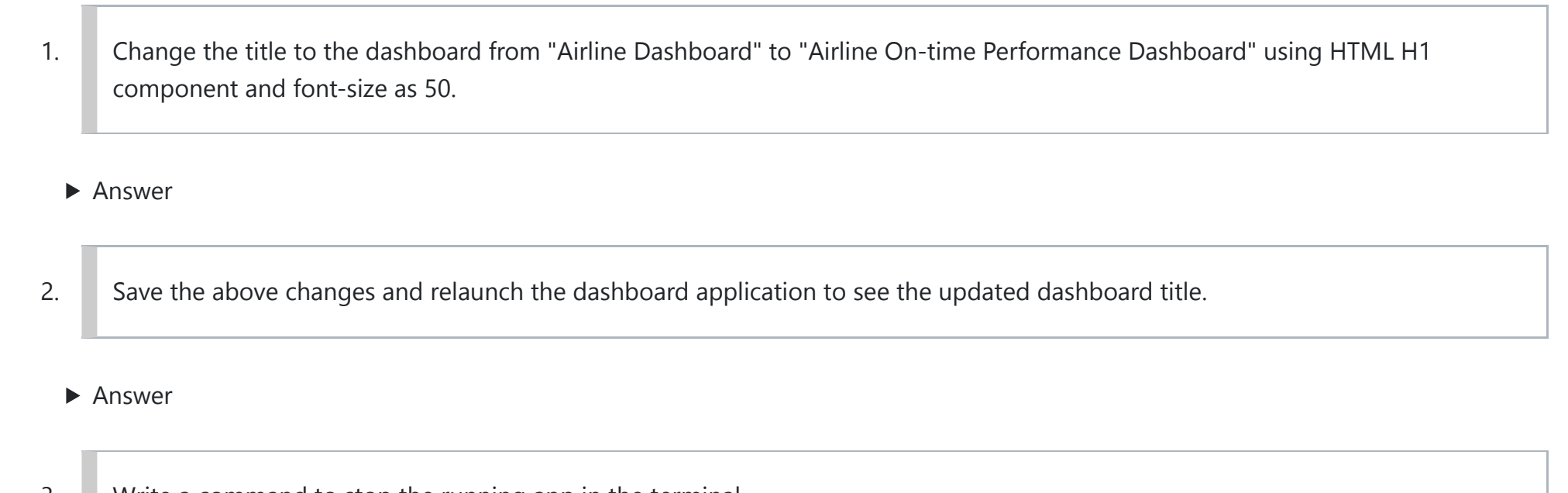
- Observe the port number shown in the terminal.



- Click on the `Launch Application` option from the side 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!

Exercise : Practice Tasks

You will practice some tasks to update the dashboard.

1. 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

2. Save the above changes and relaunch the dashboard application to see the updated dashboard title.

▶ Answer

3. Write a command to stop the running app in the terminal

▶ Answer

Author

Saishruthi Swaminathan

Değişiklik günlüğü

Tarih	Sürüm	Tarafından değiştirildi	Açıklamayı Değiştir
05-07-2021	1.1	Saishruthi	İlk sürüm oluşturuldu
24-08-2022	1.2	pratiksha	Güncellenmiş talimatlar
29-08-2022	1.3	Pratiksha Verma	Güncellenmiş Ekran Görüntüsü