



IBM Developer SKILLS NETWORK

Basic Plotly Charts

Estimated time needed: 30 minutes

Objectives

In this lab, you will learn about creating plotly charts using `plotly.graph_objects` and `plotly.express`.

Learn more about:

- [Plotly python](#)
- [Plotly Graph Objects](#)
- [Plotly Express](#)
- Handling data using [Pandas](#)

We will be using the [airline dataset](#) from [Data Asset eXchange](#).

Airline Reporting Carrier On-Time Performance Dataset

The Reporting Carrier On-Time Performance Dataset contains information on approximately 200 million domestic US flights reported to the United States Bureau of Transportation Statistics. The dataset contains basic information about each flight (such as date, time, departure airport, arrival airport) and, if applicable, the amount of time the flight was delayed and information about the reason for the delay. This dataset can be used to predict the likelihood of a flight arriving on time.

Preview data, dataset metadata, and data glossary [here](#).

```
In [1]: # Import required Libraries
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
```

Read Data

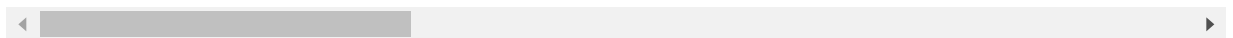
```
In [2]: # Read the airline data into pandas dataframe
airline_data = pd.read_csv('https://cf-courses-data.s3.us.cloud-object-storage.appd
                        encoding = "ISO-8859-1",
                        dtype={'Div1Airport': str, 'Div1TailNum': str,
                              'Div2Airport': str, 'Div2TailNum': str})
```

```
In [3]: # Preview the first 5 lines of the loaded data
airline_data.head()
```

```
Out[3]:
```

	Unnamed: 0	Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate	Reporting_Airline	DOT
0	1295781	1998	2	4	2	4	1998-04-02	AS	
1	1125375	2013	2	5	13	1	2013-05-13	EV	
2	118824	1993	3	9	25	6	1993-09-25	UA	
3	634825	1994	4	11	12	6	1994-11-12	HP	
4	1888125	2017	3	8	17	4	2017-08-17	UA	

5 rows × 110 columns



```
In [4]: # Shape of the data
airline_data.shape
```

```
Out[4]: (27000, 110)
```

```
In [5]: # Randomly sample 500 data points. Setting the random state to be 42 so that we get
data = airline_data.sample(n=500, random_state=42)
```

```
In [6]: # Get the shape of the trimmed data
data.shape
```

```
Out[6]: (500, 110)
```

Lab structure

plotly.graph_objects

1. Review scatter plot creation

Theme: How departure time changes with respect to airport distance

2. **To do** - Create line plot

Theme: Extract average monthly delay time and see how it changes over the year

plotly.express

1. Review bar chart creation

Theme: Extract number of flights from a specific airline that goes to a destination

2. **To do** - Create bubble chart

Theme: Get number of flights as per reporting airline

3. **To do** - Create histogram

Theme: Get distribution of arrival delay

4. Review pie chart

Theme: Proportion of distance group by month (month indicated by numbers)

5. **To do** - Create sunburst chart

Theme: Hierarchical view in othe order of month and destination state holding value of number of flights

plotly.graph_objects¶

1. Scatter Plot

Learn more about usage of scatter plot [here](#)

Idea: How departure time changes with respect to airport distance

```
In [7]: # First we create a figure using go.Figure and adding trace to it through go.scatter
fig = go.Figure(data=go.Scatter(x=data['Distance'], y=data['DepTime'], mode='markers'))
# Updating layout through `update_layout`. Here we are adding title to the plot and
fig.update_layout(title='Distance vs Departure Time', xaxis_title='Distance', yaxis_
# Display the figure
fig.show()
```

2. Line Plot

Learn more about line plot [here](#)

Idea: Extract average monthly arrival delay time and see how it changes over the year.

```
In [8]: # Group the data by Month and compute average over arrival delay time.
line_data = data.groupby('Month')['ArrDelay'].mean().reset_index()
```

```
In [9]: # Display the data
line_data
```

```
Out[9]:
```

	Month	ArrDelay
0	1	2.232558
1	2	2.687500
2	3	10.868421
3	4	6.229167
4	5	-0.279070
5	6	17.310345
6	7	5.088889
7	8	3.121951
8	9	9.081081
9	10	1.200000
10	11	-3.975000
11	12	3.240741

To do:

- Create a line plot with x-axis being the month and y-axis being computed average delay time. Update plot title, xaxis, and yaxis title.
- Hint: Scatter and line plot vary by updating mode parameter.

```
In [13]: fig = go.Figure(data=go.Scatter(x=line_data['Month'], y=line_data['ArrDelay'], mode=
fig.update_layout(title='Month vs Average Flight Delay Time', xaxis_title='Month', y
fig.show())
```

Double-click [here](#) for the solution.

plotly.express

1. Bar Chart

Learn more about bar chart [here](#)

Idea: Extract number of flights from a specific airline that goes to a destination

```
In [17]: # Group the data by destination state and reporting airline. Compute total number of  
bar_data = data.groupby('DestState')['Flights'].sum().reset_index()
```

```
In [41]: # Display the data  
bar_data
```

```
Out[41]:
```

	DestState	Flights
0	AK	4.0
1	AL	3.0
2	AZ	8.0
3	CA	68.0
4	CO	20.0
5	CT	5.0
6	FL	32.0
7	GA	27.0
8	HI	5.0
9	IA	1.0
10	ID	1.0
11	IL	33.0
12	IN	6.0
13	KS	1.0
14	KY	14.0
15	LA	4.0
16	MA	10.0
17	MD	7.0
18	MI	16.0

	DestState	Flights
19	MN	11.0
20	MO	18.0
21	MT	3.0
22	NC	13.0
23	NE	2.0
24	NH	1.0
25	NJ	5.0
26	NM	1.0
27	NV	13.0
28	NY	21.0
29	OH	9.0
30	OK	6.0
31	OR	3.0
32	PA	14.0
33	PR	2.0
34	RI	1.0
35	SC	1.0
36	TN	14.0
37	TX	60.0
38	UT	7.0
39	VA	11.0
40	VI	1.0
41	WA	10.0
42	WI	8.0

```
In [19]: # Use plotly express bar chart function px.bar. Provide input data, x and y axis var
# This will give total number of flights to the destination state.
fig = px.bar(bar_data, x="DestState", y="Flights", title='Total number of flights to
fig.show()
```

2. Bubble Chart

Learn more about bubble chart [here](#)

Idea: Get number of flights as per reporting airline

```
In [20]: # Group the data by reporting airline and get number of flights  
bub_data = data.groupby('Reporting_Airline')['Flights'].sum().reset_index()
```

```
In [43]: bub_data
```

```
Out[43]:
```

	Reporting_Airline	Flights
0	9E	5.0
1	AA	57.0
2	AS	14.0
3	B6	10.0
4	CO	12.0
5	DL	66.0
6	EA	4.0
7	EV	11.0
8	F9	4.0
9	FL	3.0
10	HA	3.0
11	HP	7.0
12	KH	1.0
13	MQ	27.0
14	NK	3.0
15	NW	26.0
16	OH	8.0
17	OO	28.0
18	PA (1)	1.0
19	PI	1.0
20	PS	1.0
21	TW	14.0
22	UA	51.0
23	US	43.0
24	VX	1.0
25	WN	86.0
26	XE	6.0
27	YV	6.0
28	YX	1.0

To do

- Create a bubble chart using the `bub_data` with x-axis being reporting airline and y-axis being flights.
- Provide title to the chart
- Update size of the bubble based on the number of flights. Use `size` parameter.
- Update name of the hover tooltip to `reporting_airline` using `hover_name` parameter.

```
In [23]: fig = px.scatter(bub_data, x='Reporting_Airline', y='Flights', size='Flights',  
                    hover_name='Reporting_Airline', title='Reporting Airline vs Number of Flights')  
fig.show()
```

Double-click **here** for the solution.

Histogram

Learn more about histogram [here](#)

Idea: Get distribution of arrival delay

```
In [24]: # Set missing values to 0  
data['ArrDelay'] = data['ArrDelay'].fillna(0)
```

To do

- Use `px.histogram` and pass the dataset.
- Pass `ArrDelay` to x parameter.

```
In [26]: fig= px.histogram(data, x='ArrDelay')  
fig.show()
```

Double-click **here** for the solution.

Pie Chart

Learn more about pie chart [here](#)

Idea: Proportion of distance group by month (month indicated by numbers)

```
In [28]: # Use px.pie function to create the chart. Input dataset.
```



```
# Values parameter will set values associated to the sector. 'Month' feature is pass
# Labels for the sector are passed to the `names` parameter.
fig = px.pie(data, values='Month', names='DistanceGroup', title='Distance group prop
fig.show()
```

Sunburst Charts

Learn more about sunburst chart [here](#)

Idea: Hierarchical view in othe order of month and destination state holding value of number of flights

To do

- Create sunburst chart using `px.sunburst` .
- Define hierarchy of sectors from root to leaves in `path` parameter. Here, we go from `Month` to `DestStateName` feature.
- Set sector values in `values` paramter. Here, we can pass in `Flights` feature.
- Show the figure.

In [29]:

```
fig = px.sunburst(data, path=['Month', 'DestStateName'], values='Flights')
fig.show()
```

Double-click **here** for the solution.

Summary

Congratulations for completing your first lab.

In this lab, you have learnt how to use `plotly.graph_objects` and `plotly.express` for creating plots and charts.

Author

[Saishruthi Swaminathan](#)

Changelog

Date	Version	Changed by	Change Description
12-18-2020	1.0	Nayef	Added dataset link and upload to Git

In []: