**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](https://plotly.com/python/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)
* [Plotly Graph Objects](https://plotly.com/python/graph-objects/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)
* [Plotly Express](https://plotly.com/python/plotly-express/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)
* Handling data using [Pandas](https://pandas.pydata.org/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)

We will be using the [airline dataset](https://developer.ibm.com/exchanges/data/all/airline/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01) from [Data Asset eXchange](https://developer.ibm.com/exchanges/data/).

**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.](https://dax-cdn.cdn.appdomain.cloud/dax-airline/1.0.1/data-preview/index.html)

[ ]:



*# Import required libraries*

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

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

**import** plotly.graph\_objects **as** go

**Read Data**

[ ]:



*# 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})

[ ]:



*# Preview the first 5 lines of the loaded data*

airline\_data.head()

[ ]:



*# Shape of the data*

airline\_data.shape

[ ]:



*# 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)

[ ]:



*# Get the shape of the trimmed data*

data.shape

**Lab structure**

**plotly.graph\_objects**

1. Review scatter plot creation

Theme: How departure time changes with respect to airport distance

1. **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

1. **To do** - Create bubble chart

Theme: Get number of flights as per reporting airline

1. **To do** - Create histogram

Theme: Get distribution of arrival delay

1. Review pie chart

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

1. **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](https://plotly.com/python/line-and-scatter/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)

**Idea: How departure time changes with respect to airport distance**

[ ]:



*# 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', marker**=**dict(color**=**'red')))

*# Updating layout through `update\_layout`. Here we are adding title to the plot and providing title to x and y axis.*

fig.update\_layout(title**=**'Distance vs Departure Time', xaxis\_title**=**'Distance', yaxis\_title**=**'DepTime')

*# Display the figure*

fig.show()

**2. Line Plot**

Learn more about line plot [here](https://plotly.com/python/line-charts/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)

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

[ ]:



*# Group the data by Month and compute average over arrival delay time.*

line\_data **=** data.groupby('Month')['ArrDelay'].mean().reset\_index()

[ ]:



*# Display the data*

line\_data

**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.

[ ]:



*# Create line plot here*

​



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

​

*<!-- The answer is below:*

​

*fig = go.Figure(data=go.Scatter(x=line\_data['Month'], y=line\_data['ArrDelay'], mode='lines', marker=dict(color='green')))*

*fig.update\_layout(title='Month vs Average Flight Delay Time', xaxis\_title='Month', yaxis\_title='ArrDelay')*

*fig.show()*

​

*-->*

​

**plotly.express¶**

**1. Bar Chart**

Learn more about bar chart [here](https://plotly.com/python/bar-charts/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)

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

[ ]:



*# Group the data by destination state and reporting airline. Compute total number of flights in each combination*

bar\_data **=** data.groupby(['DestState'])['Flights'].sum().reset\_index()

[ ]:



*# Display the data*

bar\_data

[ ]:



*# Use plotly express bar chart function px.bar. Provide input data, x and y axis variable, and title of the chart.*

*# 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 the destination state split by reporting airline')

fig.show()

**2. Bubble Chart**

Learn more about bubble chart [here](https://plotly.com/python/bubble-charts/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)

**Idea: Get number of flights as per reporting airline**

[ ]:



*# Group the data by reporting airline and get number of flights*

bub\_data **=** data.groupby('Reporting\_Airline')['Flights'].sum().reset\_index()

[ ]:



bub\_data

**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.

[ ]:



*# Create bubble chart here*

​



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

​

*<!-- The answer is below:*

​

*fig = px.scatter(bub\_data, x="Reporting\_Airline", y="Flights", size="Flights",*

*hover\_name="Reporting\_Airline", title='Reporting Airline vs Number of Flights', size\_max=60)*

*fig.show()*

​

*-->*

​

**Histogram**

Learn more about histogram [here](https://plotly.com/python/histograms/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)

**Idea: Get distribution of arrival delay**

[ ]:



*# 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.

[ ]:



*# Create histogram here*

​



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

​

*<!-- The answer is below:*

​

*fig = px.histogram(data, x="ArrDelay")*

*fig.show()*

​

*-->*

​

**Pie Chart**

Learn more about pie chart [here](https://plotly.com/python/pie-charts/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)

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

[ ]:



*# Use px.pie function to create the chart. Input dataset.*

*# Values parameter will set values associated to the sector. 'Month' feature is passed to it.*

*# labels for the sector are passed to the `names` parameter.*

fig **=** px.pie(data, values**=**'Month', names**=**'DistanceGroup', title**=**'Distance group proportion by month')

fig.show()

**Sunburst Charts**

Learn more about sunburst chart [here](https://plotly.com/python/sunburst-charts/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)

**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.

[ ]:



*# Create sunburst chart here*



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

​

*<!-- The answer is below:*

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

*fig.show()*

​

*-->*

​

**Summary**

Congratulations for completing your first lab.