Basic Plotly Charts

Plotly is a versatile, interactive, and open-source graphing library. It provides a wide range of chart types, including 2D, 3D, and statistical graphs, as well as maps and financial charts. Plotly is particularly well-suited for creating publication-quality visualizations that can be embedded in web applications or exported as standalone HTML files.

Plotly graph objects and Plotly express libraries to plot different types of charts

Plotly Libraries

plotly.graph_objects: This is a low level interface to figures, traces and layout. The Plotly graph objects module provides an automatically generated hierarchy of classes (figures, traces, and layout) called graph objects. These graph objects represent figures with a top-level class plotly.graph_objects.Figure.

plotly.express: Plotly express is a high-level wrapper for Plotly. It is a recommended starting point for creating the most common figures provided by Plotly using a simpler syntax. It uses graph objects internally. Now let us use these libraries to plot some charts We will start with plotly_graph_objects to plot line and scatter plots

> Note: You can hover the mouse over the charts whenever you want to view any statistics in the visualization charts

Exercise I: Get Started with Different Chart types in Plotly

In [3]:
```
!pip install plotly
!pip install nbformat
```

```
Requirement already satisfied: plotly in c:\users\hp omen\appdata\local\programs
\python\python311\lib\site-packages (6.1.2)
Requirement already satisfied: narwhals>=1.15.1 in c:\users\hp omen\appdata\local
\programs\python\python311\lib\site-packages (from plotly) (1.42.0)
Requirement already satisfied: packaging in c:\users\hp omen\appdata\local\progra
ms\python\python311\lib\site-packages (from plotly) (24.2)
Requirement already satisfied: nbformat in c:\users\hp omen\appdata\local\program
s\python\python311\lib\site-packages (5.10.4)
Requirement already satisfied: fastjsonschema>=2.15 in c:\users\hp omen\appdata\l
ocal\programs\python\python311\lib\site-packages (from nbformat) (2.21.1)
Requirement already satisfied: jsonschema>=2.6 in c:\users\hp omen\appdata\local
\programs\python\python311\lib\site-packages (from nbformat) (4.23.0)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in c:\users\hp omen\app
data\local\programs\python\python311\lib\site-packages (from nbformat) (5.7.2)
Requirement already satisfied: traitlets>=5.1 in c:\users\hp omen\appdata\local\p
rograms\python\python311\lib\site-packages (from nbformat) (5.14.3)
Requirement already satisfied: attrs>=22.2.0 in c:\users\hp omen\appdata\local\pr
ograms\python\python311\lib\site-packages (from jsonschema>=2.6->nbformat) (24.2.
0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in c:\users\h
p omen\appdata\local\programs\python\python311\lib\site-packages (from jsonschema
>=2.6->nbformat) (2024.10.1)
Requirement already satisfied: referencing>=0.28.4 in c:\users\hp omen\appdata\lo
cal\programs\python\python311\lib\site-packages (from jsonschema>=2.6->nbformat)
(0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\hp omen\appdata\local\p
rograms\python\python311\lib\site-packages (from jsonschema>=2.6->nbformat) (0.2
2.3)
Requirement already satisfied: platformdirs>=2.5 in c:\users\hp omen\appdata\loca
l\programs\python\python311\lib\site-packages (from jupyter-core!=5.0.*,>=4.12->n
bformat) (4.3.6)
Requirement already satisfied: pywin32>=300 in c:\users\hp omen\appdata\local\pro
grams\python\python311\lib\site-packages (from jupyter-core!=5.0.*,>=4.12->nbform
at) (308)
```

In [4]:
```python
# Required libraries
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
```

1. Scatter Plot:

A scatter plot shows the relationship between 2 variables on the x and y-axis. The data points here appear scattered when plotted on a two-dimensional plane. Using scatter plots, we can create exciting visualizations to express various relationships, such as:

- Height vs weight of persons
- Engine size vs automobile price
- Exercise time vs Body Fat

In [5]:
```python
# Let's illustrate the income VS age of people in a scatter plot
age_array = np.random.randint(25, 55, 60)

# define an array containing salesmount values
income_array = np.random.randint(300000, 700000, 3000000)
```

In [6]:
```
fig = go.Figure()
fig
```

Next we will create a scatter plot by using the add_trace function and use the go.scatter() function within it

In go.Scatter we define the x-axis data,y-axis data and define the mode as markers with color of the marker as blue

In [7]:
```
fig.add_trace(go.Scatter(x=age_array, y=income_array, mode='markers', marker=dic
```

However in the previous output title, x-axis and y-axis labels are missing. Let us use the update_layout function to update the title and labels.

In [8]:
```
fig.update_layout(title='Economic Survey', xaxis_title='Age', yaxis_title='Incom
fig.show()
```

Inferences:

From the above plot we find that the Income of a person is not correlated with age. We find that as the age increases the income may or not decrease.

2. Line Plot:

A line plot shows information that changes continuously with time. Here the data points are connected by straight lines. Line plots are also plotted on a two dimensional plane like scatter plots. Using line plots, we can create exciting visualizations to illustrate:

- Annual revenue growth
- Stock Market analysis over time
- Product Sales over time

Let us illustrate the sales of bicycles from Jan to August last year using a line chart

In [9]:
```
# Define an array containing number_of_bicyclessold
number_of_bicylessold_array = [50, 100, 40, 150, 160, 70, 60, 45]

# Define an array containing months
months_array = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug']
```

In [10]:
```
# First we will create an empty Figure using go.Figure()
fig = go.Figure()
fig.show()
```

Next we will create a line plot by using the add_trace function and use the go.scatter() function within it

In go.Scatter we define the x-axis data,y-axis data and define the mode as lines with color of the marker as green

In [11]:
```
fig.add_trace(go.Scatter(x=months_array, y=number_of_bicylessold_array, mode='li
```

Inferences:

From the above plot we find that the sales is the highest in the month of May and then there is a decline in sales.

We will now use Plotly express library to plot the other graphs

3.Bar Plot:

A bar plot represents categorical data in rectangular bars. Each category is defined on one axis, and the value counts for this category are represented on another axis. Bar charts are generally used to compare values.We can use bar plots in visualizing:

- Pizza delivery time in peak and non peak hours
- Population comparison by gender
- Number of views by movie name

Example: Let us illustrate the average pass percentage of classes from grade 6 to grade 10

In [12]:
```
# Define an array containing scores of students
score_array = [80, 90, 56, 88, 95]

# Define an array containing grade names
grade_array = ['Grade 6', 'Grade 7', 'Grade 8', 'Grade 9', 'Grade 10']
```

In plotly express we set the axis values and the title within the same function call `px.<graphtype>(x=<xaxis value source>,y=<y-axis value source>,title=<appropriate title as a string>)` .In the below code we use `px.bar( x=grade_array, y=score_array, title='Pass Percentage of Classes')` .

In [13]:
```
# Use plotly express bar chart function px.bar.Provide input data, x and y axis
# This will give average pass percentage per class
fig = px.bar( x=grade_array, y=score_array, title='Pass Percentage of Classes')
fig.show()
```

Inferences:

From the above plot we find that Grade 8 has the lowest pass percentage and Grade 10 has the highest pass percentage

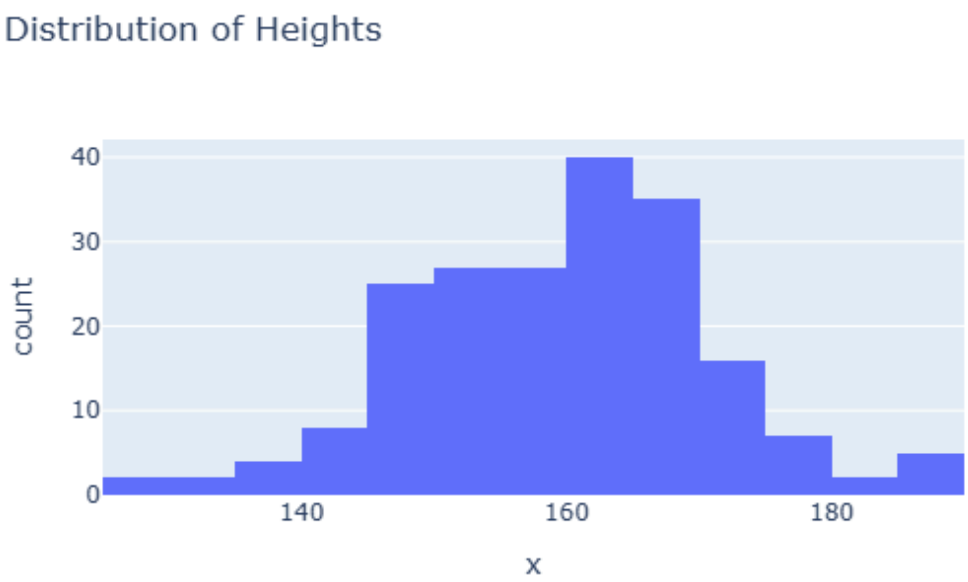
4.Histogram:

A histogram is used to represent continuous data in the form of bar. Each bar has discrete values in bar graphs, whereas in histograms, we have bars representing a range of values. Histograms show frequency distributions. We can use histograms to visualize:

- Students marks distribution
- Frequency of waiting time of customers in a Bank

In [14]:
```
##Example 4: Let us illustrate the distribution of heights of 200 people using a

import numpy as np
#Here we will concentrate on heights which are 160 and the standard deviation is
heights_array = np.random.normal(160, 11, 200)
## Use plotly express histogram chart function px.histogram.Provide input data x
fig = px.histogram(x=heights_array,title="Distribution of Heights")
fig.show()
```



Inferences

From this we can analyze that there are around 2 people who are at the height of 130cm and 45 people at the height of 160 cm

5. Bubble Plot:

A bubble plot is used to show the relationship between 3 or more variables. It is an extension of a scatter plot. Bubble plots are ideal for visualizing:

- Global Economic position of Industries
- Impact of viruses on Diseases

Example : Let us illustrate crime statistics of US cities with a bubble chart

In [15]:
```
# Create a dictionary having city, numberofcrimes and year as a 3 keys

crime_details = {
    'City' : ['Chicago', 'Chicago', 'Austin', 'Austin', 'Seattle', 'Seattle'],
    'Numberofcrimes' : [1000, 1200, 400, 700, 350, 1500],
    'Year' : ['2007', '2008', '2007', '2008', '2007', '2008'],
}

df = pd.DataFrame(crime_details)
df
```

Out[15]:

	City	Numberofcrimes	Year
0	Chicago	1000	2007
1	Chicago	1200	2008
2	Austin	400	2007
3	Austin	700	2008
4	Seattle	350	2007
5	Seattle	1500	2008

In [16]:
```
# Group the number of crimes by city and find the total number of crimes per cit
bub_data = df.groupby('City')['Numberofcrimes'].sum().reset_index()
```

In [17]:
```
bub_data
```

Out[17]:

	City	Numberofcrimes
0	Austin	1100
1	Chicago	2200
2	Seattle	1850

In [18]:
```
# Bubble chart using px.scatter function with x ,y and size varibles defined.Tit
fig = px.scatter(bub_data, x='City', y='Numberofcrimes', size='Numberofcrimes',
                 hover_name='City', title='Crime Statistics', size_max=60)

fig.show()
```

Inferences

The size of the bubble in the bubble chart indicates that Chicago has the highest crime rate when compared with the other 2 cities.

6.Pie Plot:

A pie plot is a circle chart mainly used to represent proportion of part of given data with respect to the whole data. Each slice represents a proportion and on total of the proportion becomes a whole. We can use bar plots in visualizing:

- Sales turnover percentatge with respect to different products
- Monthly expenditure of a Family

In [19]:
```
# Monthly expenditure of a family
exp_percent = [20, 50, 10, 8, 12]
house_holdcategories = ['Grocery', 'Rent', 'School Fees', 'Transport', 'Savings'
```

In [20]:
```
fig = px.pie(values = exp_percent, names=house_holdcategories, title='Household
fig.show()
```

Inferences

From this pie chart we can find that the family expenditure is maximum for rent.

7.Sunburst Charts:

Sunburst charts represent hierarchial data in the form of concentric circles. Here the innermost circle is the root node which defines the parent, and then the outer rings move down the hierarchy from the centre. They are also called radial charts.We can use them to plot

- Worldwide mobile Sales where we can drill down as follows:
    - innermost circle represents total sales
    - first outer circle represents continentwise sales
    - second outer circle represents countrywise sales within each continent
- Disease outbreak hierarchy
- Real Estate Industrial chain

Example : Lets us illustrate plot

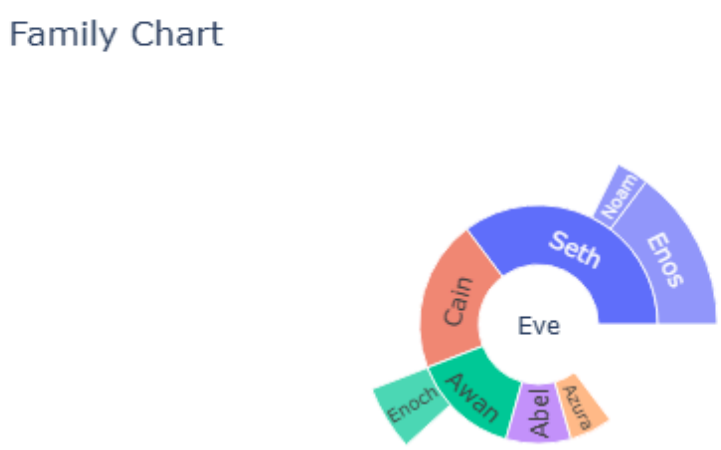
Create a dictionary having a set of people represented by a character array and the parents of these characters represented in another

array and the values are the values associated to the vectors.

In [28]:
```
data = dict(
    character = ["Eve", "Cain", "Seth", "Enos", "Noam", "Abel", "Awan", "Enoch",
    parent = ["", "Eve", "Eve", "Seth", "Seth", "Eve", "Eve", "Awan", "Eve"],
    value = [10, 14, 12, 10, 2, 6, 6, 4, 4]
)
```

```
fig = px.sunburst(
    data,
    names = 'character',
    parents = 'parent',
    values = 'value',
    title = 'Family Chart'
)

fig.show()
```



Inferences

It is found that here the innermost circle **Eve** represents the parent and the second outer circle represents his children **Cain**,**Seth** and so on.Further the outermost circle represents his grandchildren **Enoch** and **Enos**

In [ ]: