

Lab: AI & ML Lab (23A31403)

College: **Andhra**

Engineering College

2. Pandas Library: Visualization

a) Write a program which use pandas inbuilt visualization to plot following graphs:

- i. Bar plots
- ii. Histograms
- iii. Line plots
- iv. Scatter plots

dir(pd.DataFrame)

```
[  
...  
'pivot',  
'pivot_table',  
'plot',  
'pop',  
'pow',  
.....  
]
```

print(pd.DataFrame.plot.__doc__)

Make plots of Series or DataFrame.

Uses the backend specified by the option ``plotting.backend``. By default, matplotlib is used.

Parameters

data : Series or DataFrame

The object for which the method is called.

x : label or position, default None

Only used if data is a DataFrame.

y : label, position or list of label, positions, default None

Allows plotting of one column versus another. Only used if data is a DataFrame.

kind : str

The kind of plot to produce:

- 'line' : line plot (default)
- 'bar' : vertical bar plot
- 'barh' : horizontal bar plot

- 'hist' : histogram
 - 'box' : boxplot
 - 'kde' : Kernel Density Estimation plot
 - 'density' : same as 'kde'
 - 'area' : area plot
 - 'pie' : pie plot
 - 'scatter' : scatter plot (DataFrame only)
 - 'hexbin' : hexbin plot (DataFrame only)
- ax : matplotlib axes object, default None
 An axes of the current figure.
- subplots : bool or sequence of iterables, default False
 Whether to group columns into subplots:
- ``False`` : No subplots will be used
 - ``True`` : Make separate subplots for each column.
 - sequence of iterables of column labels: Create a subplot for each group of columns. For example ``(['a', 'c'), ('b', 'd')]`` will create 2 subplots: one with columns 'a' and 'c', and one with columns 'b' and 'd'. Remaining columns that aren't specified will be plotted in additional subplots (one per column).

Program :

```
import pandas as pd
dict = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
        "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing", "Pretoria"],
        "area": [8.516, 17.10, 3.286, 9.597, 1.221],
        "count01": [8.516, 17.10, 3.286, 9.597, 1.221],
        "population": [200.4, 143.5, 1252, 1357, 52.98] }
brics = pd.DataFrame(dict)

print(brics)
```

Output:

	country	capital	area	count01	population
0	Brazil	Brasilia	8.516	8.516	200.40
1	Russia	Moscow	17.100	17.100	143.50
2	India	New Dehli	3.286	3.286	1252.00
3	China	Beijing	9.597	9.597	1357.00
4	South Africa	Pretoria	1.221	1.221	52.98

Write From Here

Program:

```
# -----  
# 1. BAR PLOT (Area by Country)  
# -----  
brics.plot(kind='bar', x='country', y='area', title='Area of BRICS Countries')  
plt.ylabel("Area (million sq km)")  
plt.show()  
  
# -----  
# 2. HISTOGRAM (Population distribution)  
# -----  
brics['population'].plot(kind='hist', title='Population Histogram')  
plt.xlabel("Population (millions)")  
plt.show()  
  
# -----  
# 3. LINE PLOT (Area trend)  
# -----  
brics.plot(kind='line', x='country', y='area', title='Area Line Plot')  
plt.ylabel("Area (million sq km)")  
plt.show()  
  
# -----  
# 4. SCATTER PLOT (Area vs Population)  
# -----  
brics.plot(kind='scatter', x='area', y='population',  
          title='Scatter Plot: Area vs Population')  
plt.xlabel("Area (million sq km)")  
plt.ylabel("Population (millions)")  
plt.show()
```

Output :





