Data Visualization

It is hard to digest many data at the same time. Data visualizaiton is a way to make the process simple and straightforward.

Because Data Visualization is so important and essential in today's business, pandas as a data-driven package provides built-in plotting functions. We will learn some basic usage of them today.

Setup

```
import numpy as np
import pandas as pd
```

```
df = pd.read_csv('/content/Economy_of_US.csv')
df
```

	Year	GDP_PPP	GDP_PerCapita_PPP	GDP_Nominal	GDP_PerCapita_Nominal	GDP_G
0	1980	2857.3	12552.9	2857.3	12552.9	
1	1981	3207.0	13948.7	3207.0	13948.7	
2	1982	3343.8	14405.0	3343.8	14405.0	
3	1983	3634.0	15513.7	3634.0	15513.7	
4	1984	4037.7	17086.4	4037.7	17086.4	
5	1985	4339.0	18199.3	4339.0	18199.3	
6	1986	4579.6	19034.8	4579.6	19034.8	
7	1987	4855.3	20001.0	4855.3	20001.0	
8	1988	5236.4	21376.0	5236.4	21376.0	
9	1989	5641.6	22814.1	5641.6	22814.1	
10	1990	5963.1	23848.0	5963.1	23848.0	
11	1991	6158.1	24302.8	6158.1	24302.8	
12	1992	6520.3	25392.9	6520.3	25392.9	
13	1993	6858.6	26364.2	6858.6	26364.2	

14	1994	7287.3	27674.0	7287.3	27674.0
15	1995	7639.8	28671.5	7639.8	28671.5
16	1996	8073.1	29947.0	8073.1	29947.0
17	1997	8577.6	31440.1	8577.6	31440.1
18	1998	9062.8	32833.7	9062.8	32833.7
19	1999	9631.2	34496.2	9631.2	34496.2
20	2000	10251.0	36312.8	10251.0	36312.8
21	2001	10581.9	37101.5	10581.9	37101.5
22	2002	10929.1	37945.8	10929.1	37945.8
23	2003	11456.5	39405.4	11456.5	39405.4
24	2004	12217.2	41641.6	12217.2	41641.6
25	2005	13039.2	44034.3	13039.2	44034.3
26	2006	13815.6	46216.9	13815.6	46216.9
27	2007	14474.3	47943.4	14474.3	47943.4
28	2008	14769.9	48470.6	14769.9	48470.6
29	2009	14478.1	47102.4	14478.1	47102.4
30	2010	15049.0	48586.3	15049.0	48586.3
31	2011	15599.7	50008.1	15599.7	50008.1
32	2012	16254.0	51736.7	16254.0	51736.7
33	2013	16843.2	53245.5	16843.2	53245.5
34	2014	17550.7	55083.5	17550.7	55083.5
35	2015	18206.0	56729.7	18206.0	56729.7
36	2016	18695.1	57840.0	18695.1	57840.0
37	2017	19479.6	59885.7	19479.6	59885.7
38	2018	20527.2	62769.7	20527.2	62769.7
39	2019	21372.6	65051.9	21372.6	65051.9
40	2020	20893.8	63078.5	20893.8	63078.5
41	2021	22996.1	69227.1	22996.1	69227.1
42	2022	25035.2	75179.6	25035.2	75179.6
43	2023	26185.2	78421.9	26185.2	78421.9

44	2024	27057.2	80779.3	27057.2	80779.3
45	2025	28045.3	83463.2	28045.3	83463.2
46	2026	29165.5	86521.2	29165.5	86521.2
47	2027	30281.5	89546.4	30281.5	89546.4

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Year                   48 non-null     int64  
 1   GDP_PPP                48 non-null     float64
 2   GDP_PerCapita_PPP      48 non-null     float64
 3   GDP_Nominal            48 non-null     float64
 4   GDP_PerCapita_Nominal  48 non-null     float64
 5   GDP_Growth             48 non-null     float64
 6   Inflation              48 non-null     float64
 7   Unemployment           48 non-null     float64
 8   Inflation_Change       47 non-null     object 
dtypes: float64(7), int64(1), object(1)
memory usage: 3.5+ KB
```

```
df.describe()
```

	Year	GDP_PPP	GDP_PerCapita_PPP	GDP_Nominal	GDP_PerCapita_Nomina
count	48.00	48.000000	48.000000	48.000000	48.0000
mean	2003.50	13182.360417	43192.318750	13182.360417	43192.3187
std	14.00	7817.386178	21396.888688	7817.386178	21396.8886
min	1980.00	2857.300000	12552.900000	2857.300000	12552.9000
25%	1991.75	6429.750000	25120.375000	6429.750000	25120.3750
50%	2003.50	11836.850000	40523.500000	11836.850000	40523.5000
75%	2015.25	18328.275000	57007.275000	18328.275000	57007.2750
max	2027.00	30281.500000	89546.400000	30281.500000	89546.4000

Scatter Plots

```
df.plot(x = 'Year', y = 'Inflation', kind = 'scatter')
```

```
<Axes: xlabel='Year', ylabel='Inflation'>
```

Line Plots

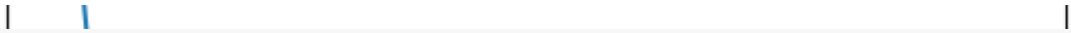

```
df.plot(y = 'Inflation', kind = 'line')
```

```
<Axes: >
```


```
df.plot(x = 'Year', y = 'Inflation', kind = 'line')
```

```
<Axes: xlabel='Year'>
```

Area Plots



```
df.plot(x = 'Year', y = 'GDP_PPP', kind = 'area')
```

```
<Axes: xlabel='Year'>
```

```
df.plot(x = 'Year', y = 'GDP_Nominal', kind = 'area')
```

```
<Axes: xlabel='Year'>
```

Bar Charts


```
df.plot(x = 'Year', y = 'GDP_PerCapita_PPP', kind = 'bar')
```

```
<Axes: xlabel='Year'>
```


```
df.plot(x = 'Year', y = 'GDP_PerCapita_Nominal', kind = 'barh')
```

```
<Axes: ylabel='Year'>
```

Histograms


```
df['Inflation'].plot(kind = 'hist')
```

```
<Axes: ylabel='Frequency'>
```


```
df['Inflation'].plot(kind = 'hist', bins = 100)
```

```
<Axes: ylabel='Frequency'>
```

```
df['Unemployment'].plot(kind = 'hist')
```

```
<Axes: ylabel='Frequency'>
```

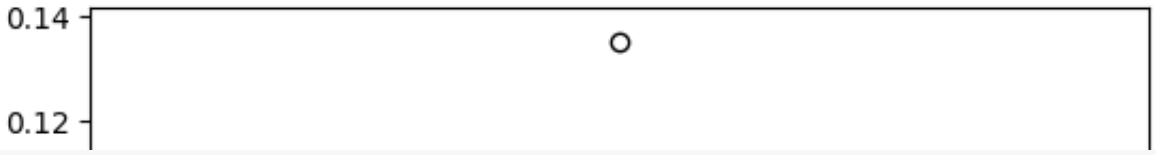

```
df['Unemployment'].plot(kind = 'kde')
```

```
<Axes: ylabel='Density'>
```

Box plot

```
df['Inflation'].plot(kind = 'box')
```

```
<Axes: >
```




```
df['Unemployment'].plot(kind = 'box')
```

```
<Axes: >
```

Pie Charts

```
df['Inflation_Change'].value_counts()
```

```
Decrease     23
Increase     21
No change     3
Name: Inflation_Change, dtype: int64
```

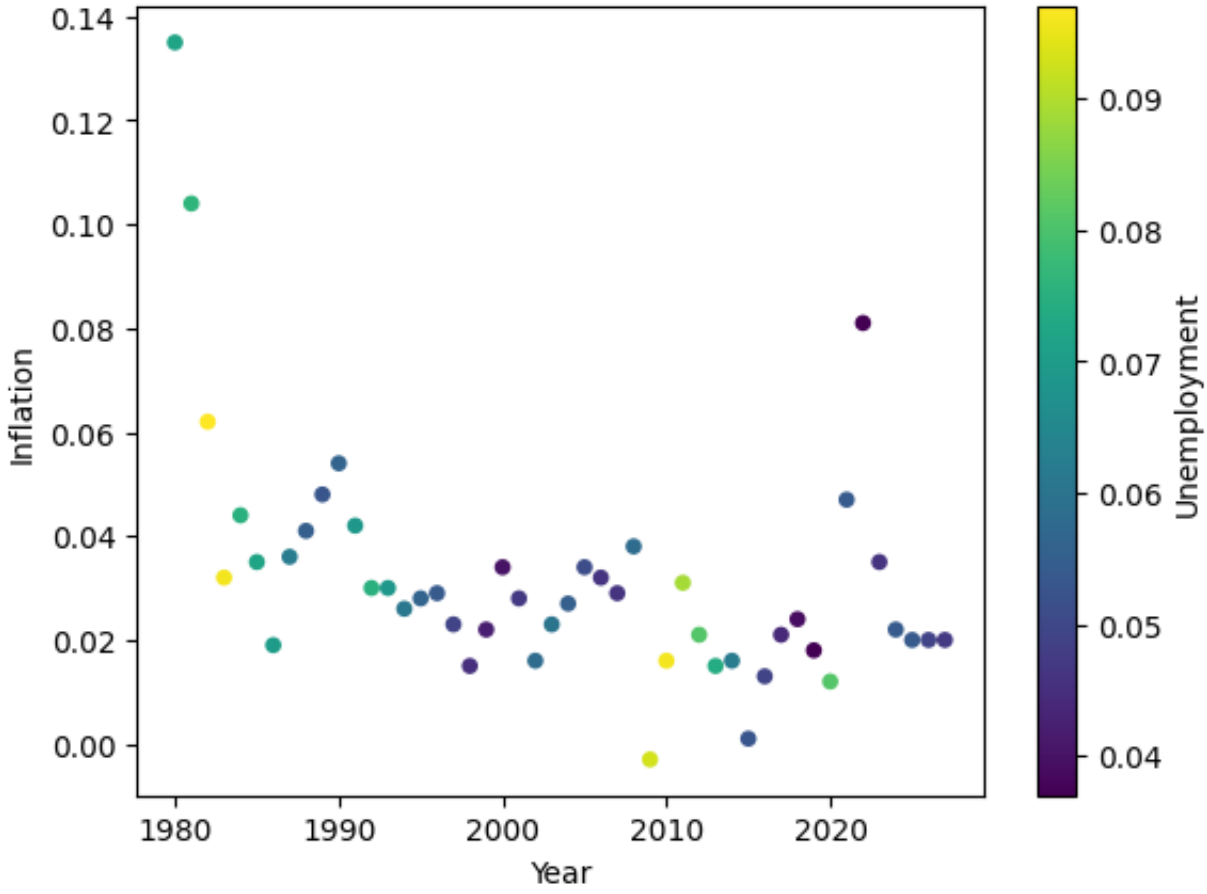
```
df['Inflation_Change'].value_counts().plot(kind = 'pie')
```

```
<Axes: ylabel='Inflation_Change'>
```

Color map

```
df.plot.scatter(x = 'Year', y = 'Inflation', c = 'Unemployment')
```

```
<Axes: xlabel='Year', ylabel='Inflation'>
```



Documentation

- You can find more details in the documentation here: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html
- Here is another useful reference: https://pandas.pydata.org/docs/user_guide/visualization.html