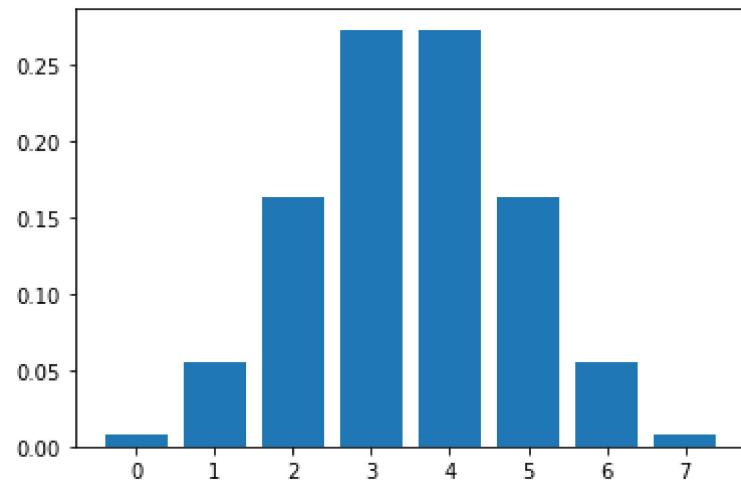


Binomial

```
In [2]: from scipy.stats import binom  
import matplotlib.pyplot as plt
```

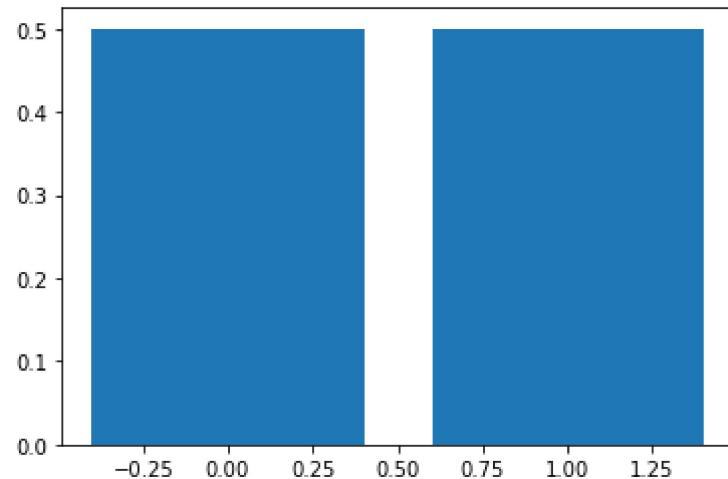
```
In [5]: n=7  
p=0.5  
r_values=list(range(n+1))  
dist=[binom.pmf(r,n,p) for r in r_values]  
plt.bar(r_values,dist)  
plt.show()
```



Bernoulli

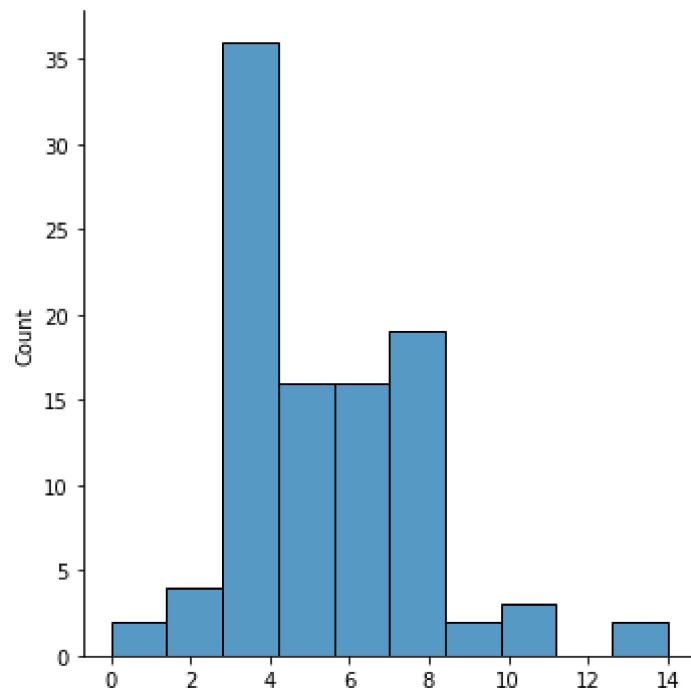
```
In [9]: import matplotlib.pyplot  
from scipy.stats import bernoulli
```

```
In [10]: bd=bernoulli(0.5)
x=[0,1]
plt.bar(x,bd.pmf(x))
plt.show()
```



poisson distribution

```
In [16]: from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns
sns.displot(random.poisson(lam=5,size=100))
plt.show()
```

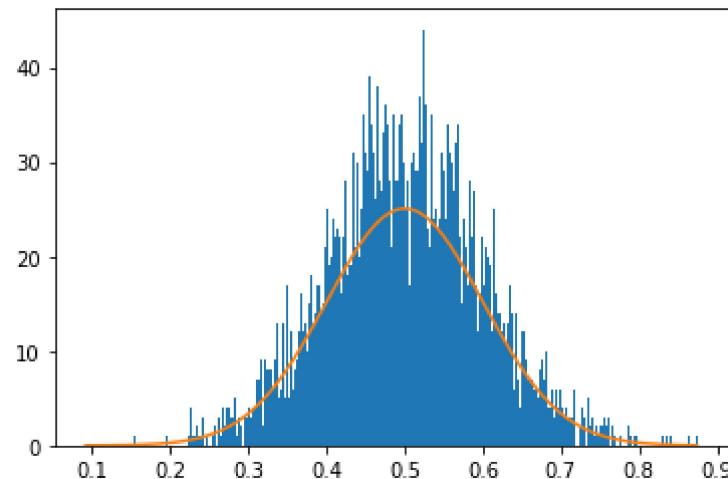


normal

```
In [15]: import matplotlib.pyplot as plt
import numpy as np

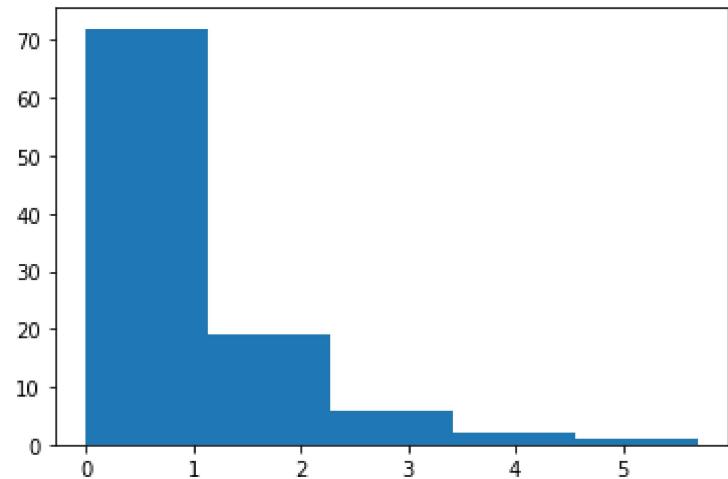
mu,sigma=0.5,0.1
s=np.random.normal(mu,sigma,1000)
count,bins,ignored=plt.hist(s,1000)

plt.plot(bins, 1/sigma*np.sqrt(2*np.pi)*np.exp(-(bins-mu)**2/(2*sigma**2)))
plt.show()
```



Exponential

```
In [12]: import numpy as np
import matplotlib.pyplot as plt
exp=np.random.exponential(1,100)
count,bins,ignored=plt.hist(exp,5)
plt.show()
```



Importing first dataset

```
In [17]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as pp
```

```
In [19]: df=pd.read_csv(r'C:\Users\user\Desktop\4_drug200.csv')
```

```
In [20]: df.head(10)
```

Out[20]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
5	22	F	NORMAL	HIGH	8.607	drugX
6	49	F	NORMAL	HIGH	16.275	drugY
7	41	M	LOW	HIGH	11.037	drugC
8	60	M	NORMAL	HIGH	15.171	drugY
9	43	M	LOW	NORMAL	19.368	drugY

```
In [21]: df.tail(11)
```

Out[21]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
189	64	M	HIGH	NORMAL	20.932	drugY
190	58	M	HIGH	HIGH	18.991	drugY
191	23	M	HIGH	HIGH	8.011	drugA
192	72	M	LOW	HIGH	16.310	drugY
193	72	M	LOW	HIGH	6.769	drugC
194	46	F	HIGH	HIGH	34.686	drugY
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

In [22]: df.describe()

Out[22]:

	Age	Na_to_K
count	200.000000	200.000000
mean	44.315000	16.084485
std	16.544315	7.223956
min	15.000000	6.269000
25%	31.000000	10.445500
50%	45.000000	13.936500
75%	58.000000	19.380000
max	74.000000	38.247000

In [23]: np.shape(df)

Out[23]: (200, 6)

In [24]: np.size(df)

Out[24]: 1200

In [25]: df.dropna()

Out[25]:

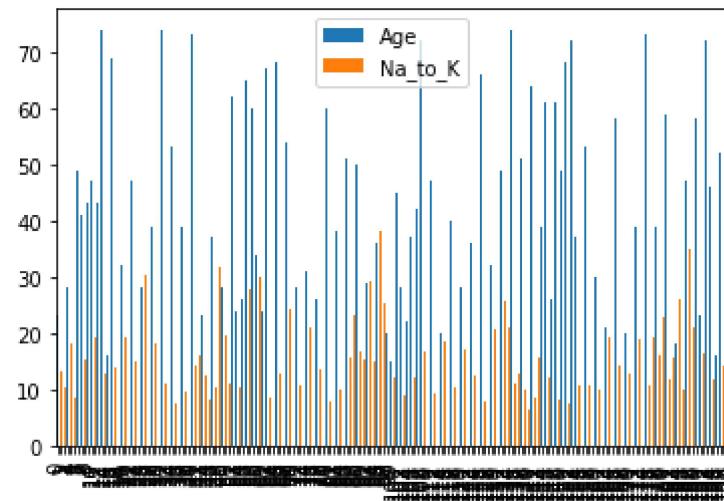
	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

Visualisation

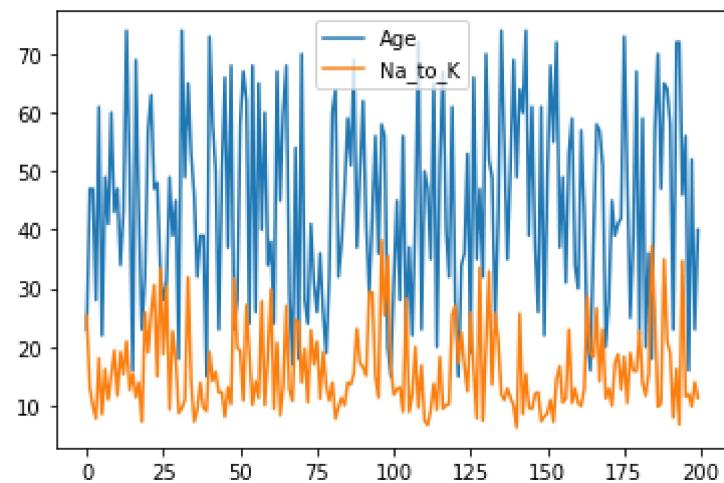
In [27]: `df.plot.bar()`

Out[27]: <AxesSubplot:>



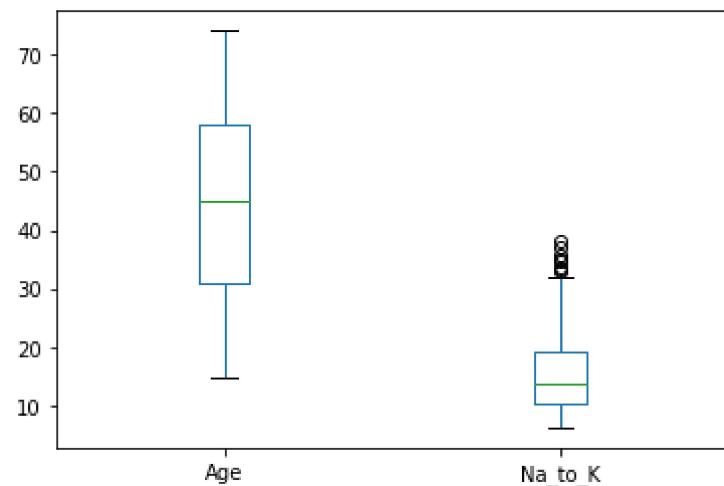
In [28]: `df.plot.line()`

Out[28]: <AxesSubplot:>



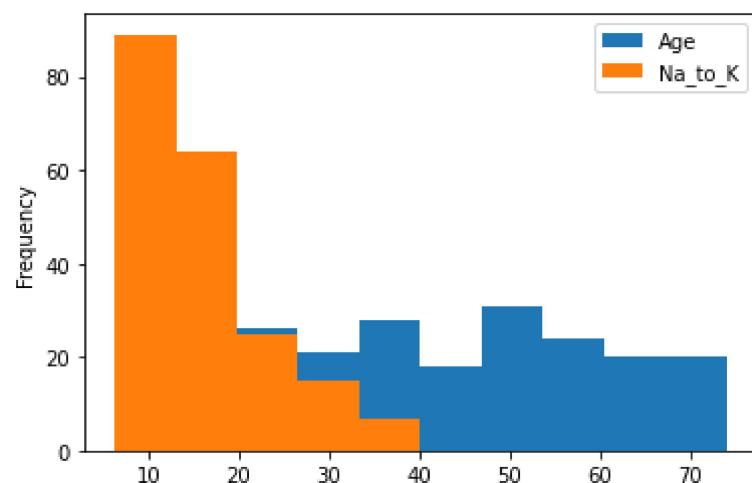
```
In [29]: df.plot.box()
```

```
Out[29]: <AxesSubplot:>
```



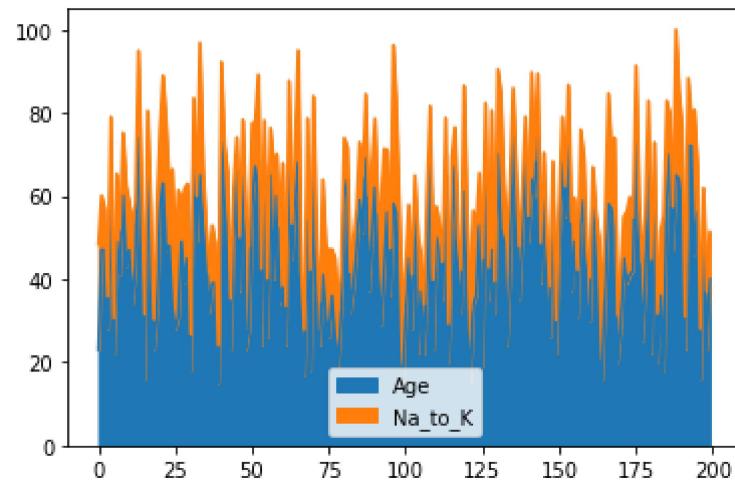
```
In [30]: df.plot.hist()
```

```
Out[30]: <AxesSubplot:ylabel='Frequency'>
```



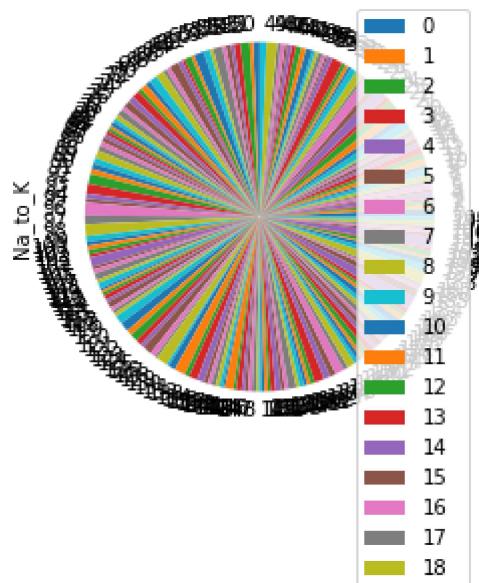
```
In [31]: df.plot.area()
```

```
Out[31]: <AxesSubplot:>
```



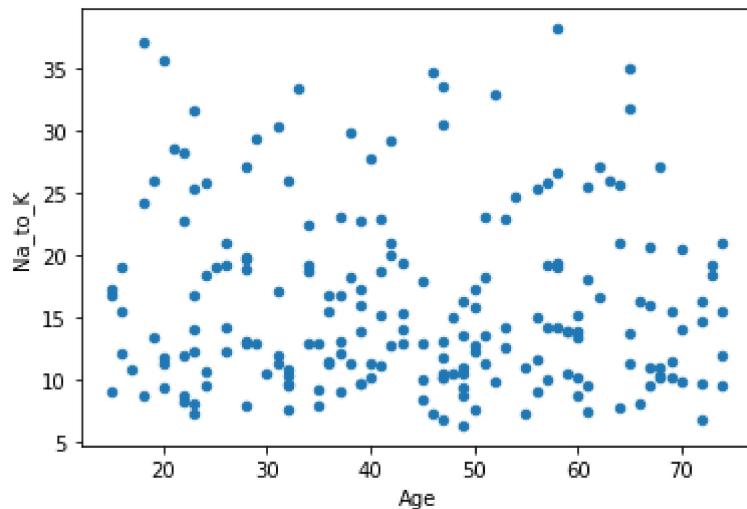
```
In [33]: df.plot.pie(y='Na_to_K')
```

```
Out[33]: <AxesSubplot:ylabel='Na_to_K'>
```



```
In [34]: df.plot.scatter(x='Age',y='Na_to_K')
```

```
Out[34]: <AxesSubplot:xlabel='Age', ylabel='Na_to_K'>
```



Importing second dataset

```
In [35]: df=pd.read_csv(r'C:\Users\user\Desktop\5_Instagram_data.csv')
```

In [36]: df.head(10)

Out[36]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Fol
0	3920	2586	1028	619	56	98	9	5	162	35	
1	5394	2727	1838	1174	78	194	7	14	224	48	
2	4021	2085	1188	0	533	41	11	1	131	62	
3	4528	2700	621	932	73	172	10	7	213	23	
4	2518	1704	255	279	37	96	5	4	123	8	
5	3884	2046	1214	329	43	74	7	10	144	9	
6	2621	1543	599	333	25	22	5	1	76	26	
7	3541	2071	628	500	60	135	4	9	124	12	
8	3749	2384	857	248	49	155	6	8	159	36	

Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Fol
9	4115	2609	1104	178	46	122	6	3	191	31

In [37]: df.tail(11)

Out[37]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	F
108	6814	2816	2769	900	128	469	8	22	249	24	
109	17713	2449	2141	12389	561	504	3	23	308	70	
110	5563	3813	362	1135	76	149	5	8	163	22	
111	4842	1658	694	2036	310	55	6	4	86	46	
112	11149	4439	747	5762	53	273	4	13	210	61	
113	10206	2371	1624	6000	117	182	10	17	172	237	
114	13700	5185	3041	5352	77	573	2	38	373	73	
115	5731	1923	1368	2266	65	135	4	1	148	20	
116	4139	1133	1538	1367	33	36	0	1	92	34	
117	32695	11815	3147	17414	170	1095	2	75	549	148	
118	36919	13473	4176	16444	2547	653	5	26	443	611	



In [38]: `df.describe()`

Out[38]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments
count	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.66
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.54
min	1941.000000	1133.000000	116.000000	0.000000	9.000000	22.000000	0.00
25%	3467.000000	1945.000000	726.000000	157.500000	38.000000	65.000000	4.00
50%	4289.000000	2207.000000	1278.000000	326.000000	74.000000	109.000000	6.00
75%	6138.000000	2602.500000	2363.500000	689.500000	196.000000	169.000000	8.00
max	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	1095.000000	19.00

In [40]: `np.shape(df)`

Out[40]: (119, 13)

In [41]: `np.size(df)`

Out[41]: 1547

In [42]: `df.isna()`

Out[42]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	F
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...
114	False	False	False	False	False	False	False	False	False	False	False
115	False	False	False	False	False	False	False	False	False	False	False
116	False	False	False	False	False	False	False	False	False	False	False
117	False	False	False	False	False	False	False	False	False	False	False
118	False	False	False	False	False	False	False	False	False	False	False

119 rows × 13 columns

In [43]: `df.fillna(value='0')`

Out[43]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	F
0	3920	2586	1028	619	56	98	9	5	162	35	
1	5394	2727	1838	1174	78	194	7	14	224	48	
2	4021	2085	1188	0	533	41	11	1	131	62	
3	4528	2700	621	932	73	172	10	7	213	23	
4	2518	1704	255	279	37	96	5	4	123	8	
...
114	13700	5185	3041	5352	77	573	2	38	373	73	
115	5731	1923	1368	2266	65	135	4	1	148	20	
116	4139	1133	1538	1367	33	36	0	1	92	34	
117	32695	11815	3147	17414	170	1095	2	75	549	148	

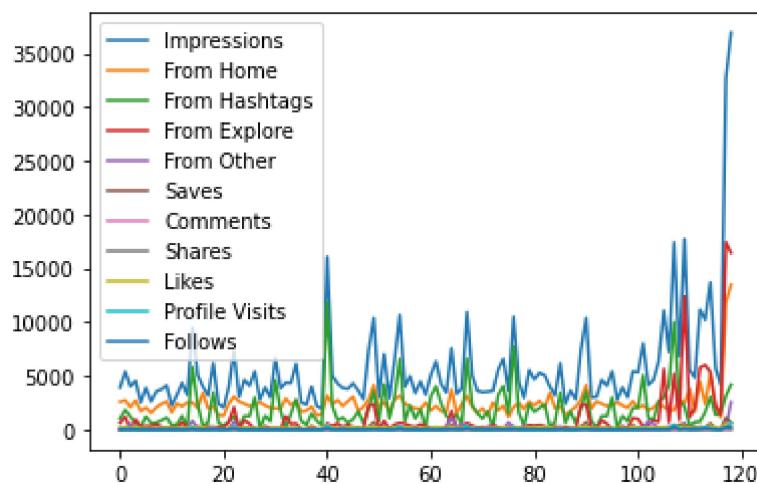
Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	F
118	36919	13473	4176	16444	2547	653	5	26	443	611

119 rows × 13 columns

Visualization

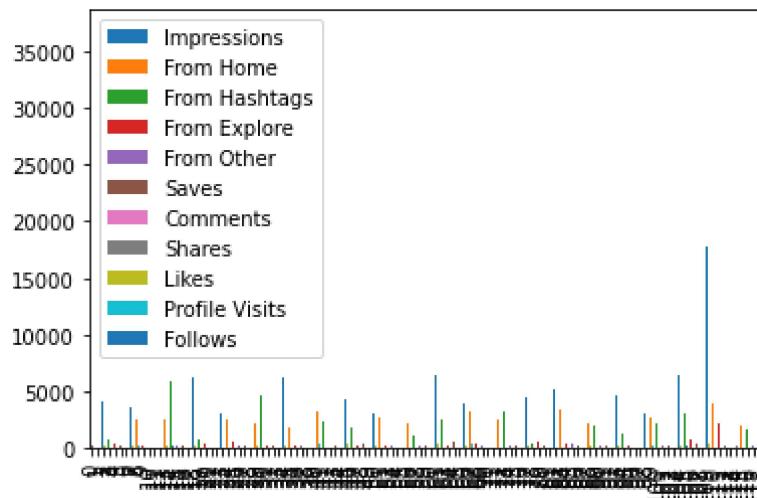
In [44]: `df.plot.line()`

Out[44]: <AxesSubplot:>



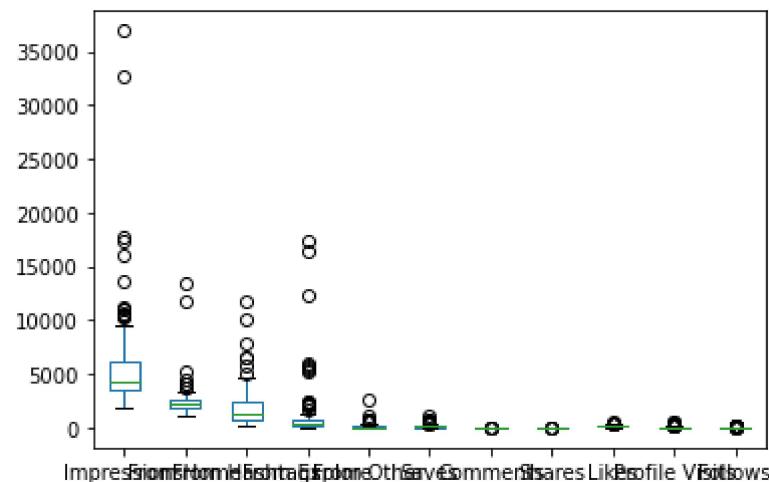
In [45]: `df.plot.bar()`

Out[45]: <AxesSubplot:>



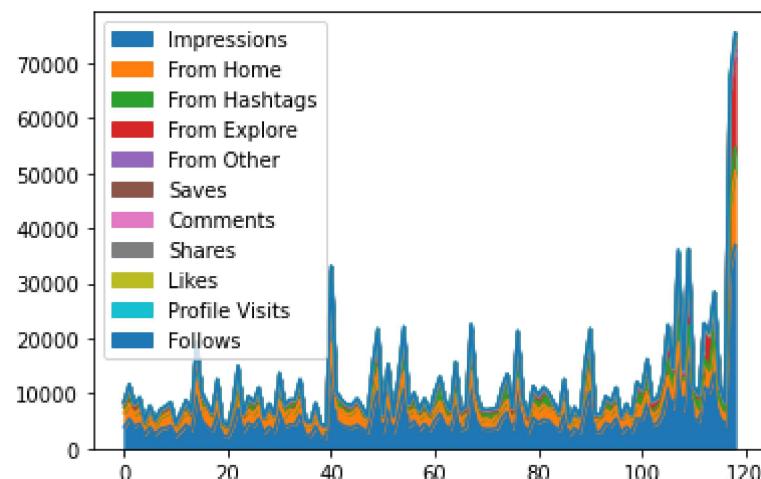
In [46]: `df.plot.box()`

Out[46]: <AxesSubplot:>



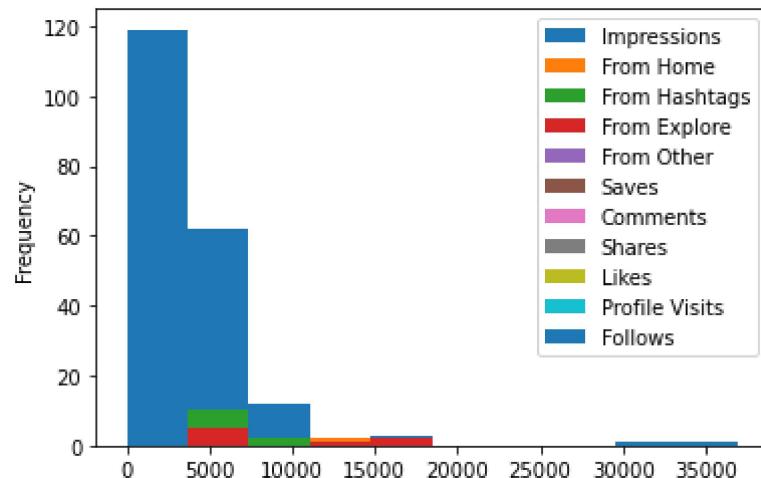
In [47]: `df.plot.area()`

Out[47]: <AxesSubplot:>



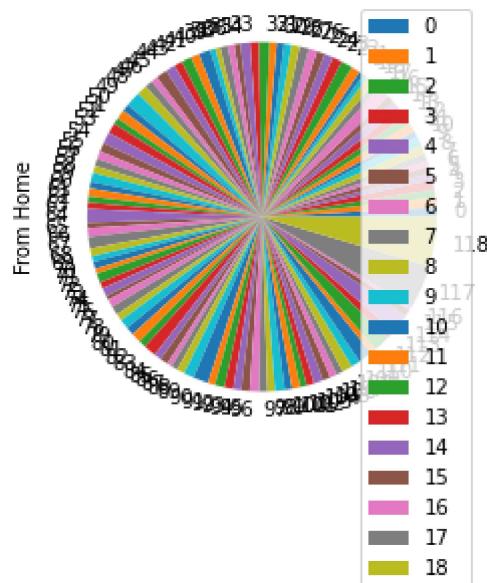
```
In [48]: df.plot.hist()
```

```
Out[48]: <AxesSubplot:ylabel='Frequency'>
```



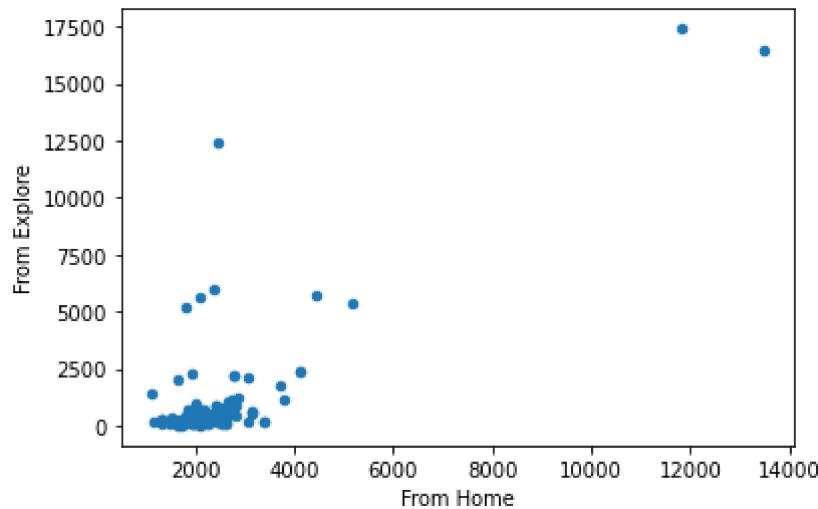
```
In [49]: df.plot.pie(y='From Home')
```

```
Out[49]: <AxesSubplot:ylabel='From Home'>
```



```
In [50]: df.plot.scatter(x='From Home',y='From Explore')
```

```
Out[50]: <AxesSubplot:xlabel='From Home', ylabel='From Explore'>
```



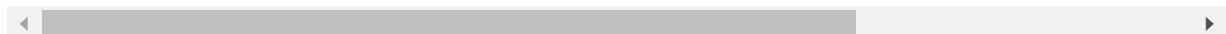
Importing third dataset

```
In [51]: df=pd.read_csv(r'C:\Users\user\Desktop\2015 dataset.csv')
```

In [52]: df.head(10)

Out[52]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63
5	Finland	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	0.64
6	Netherlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	0.61
7	Sweden	Western Europe	8	7.364	0.03157	1.33171	1.28907	0.91087	0.65
8	New Zealand	Australia and New Zealand	9	7.286	0.03371	1.25018	1.31967	0.90837	0.63
9	Australia	Australia and New Zealand	10	7.284	0.04083	1.33358	1.30923	0.93156	0.65



In [53]: df.tail(11)

Out[53]:

		Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Fre
147		Central African Republic	Sub-Saharan Africa	148	3.678	0.06112	0.07850	0.00000	0.06699	0
148		Chad	Sub-Saharan Africa	149	3.667	0.03830	0.34193	0.76062	0.15010	0
149		Guinea	Sub-Saharan Africa	150	3.656	0.03590	0.17417	0.46475	0.24009	0
150	Ivory Coast		Sub-Saharan Africa	151	3.655	0.05141	0.46534	0.77115	0.15185	0
151	Burkina Faso		Sub-Saharan Africa	152	3.587	0.04324	0.25812	0.85188	0.27125	0
152	Afghanistan		Southern Asia	153	3.575	0.03084	0.31982	0.30285	0.30335	0
153	Rwanda		Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0
154	Benin		Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0
155	Syria		Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0
156	Burundi		Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0
157	Togo		Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0



In [54]: df.describe()

Out[54]:

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	(GC)
count	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000
mean	79.493671	5.375734	0.047885	0.846137	0.991046	0.630259	0.428615	
std	45.754363	1.145010	0.017146	0.403121	0.272369	0.247078	0.150693	
min	1.000000	2.839000	0.018480	0.000000	0.000000	0.000000	0.000000	
25%	40.250000	4.526000	0.037268	0.545808	0.856823	0.439185	0.328330	
50%	79.500000	5.232500	0.043940	0.910245	1.029510	0.696705	0.435515	
75%	118.750000	6.243750	0.052300	1.158448	1.214405	0.811013	0.549092	
max	158.000000	7.587000	0.136930	1.690420	1.402230	1.025250	0.669730	

In [55]: np.shape(df)

Out[55]: (158, 12)

In [56]: np.size(df)

Out[56]: 1896

In [57]: df.isna()

Out[57]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...
153	False	False	False	False	False	False	False	False	False
154	False	False	False	False	False	False	False	False	False
155	False	False	False	False	False	False	False	False	False
156	False	False	False	False	False	False	False	False	False
157	False	False	False	False	False	False	False	False	False

158 rows × 12 columns

In [58]: df.dropna()

Out[58]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Fre
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.
...
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.

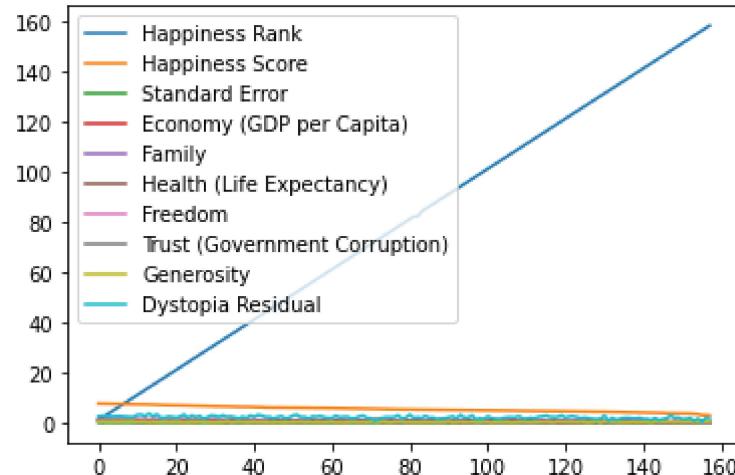
158 rows × 12 columns



Visualization

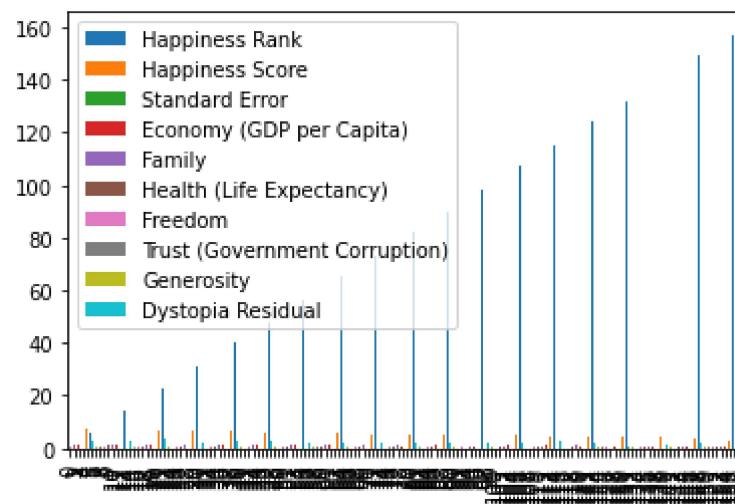
In [59]: `df.plot.line()`

Out[59]: <AxesSubplot:>



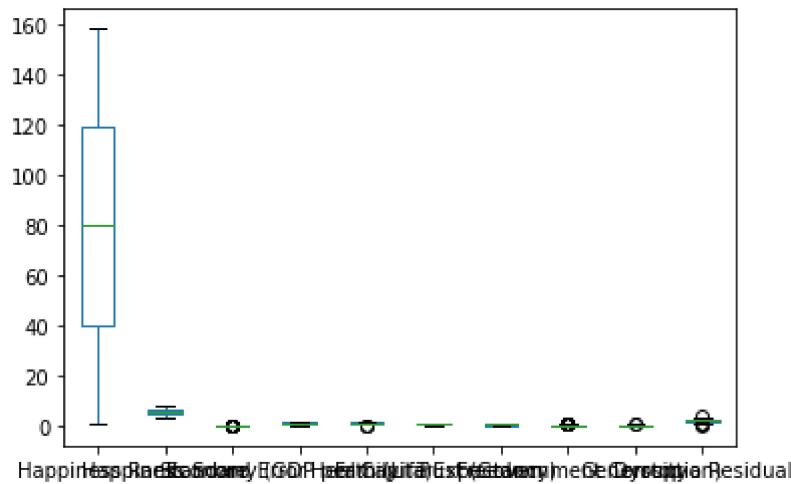
In [60]: `df.plot.bar()`

Out[60]: <AxesSubplot:>



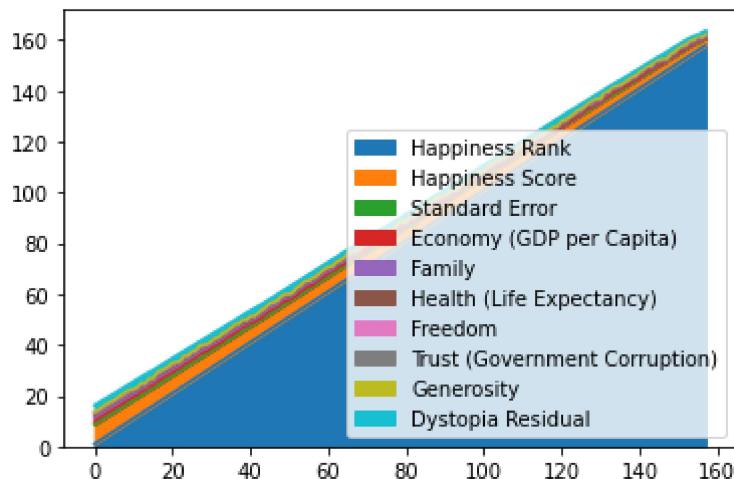
```
In [61]: df.plot.box()
```

```
Out[61]: <AxesSubplot:>
```



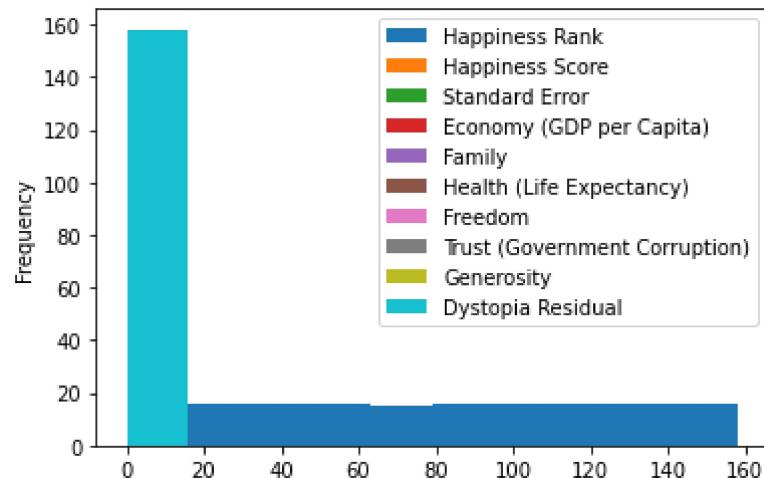
```
In [62]: df.plot.area()
```

```
Out[62]: <AxesSubplot:>
```



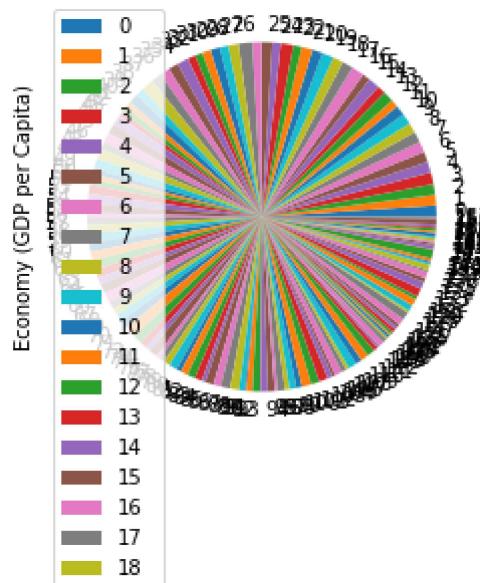
```
In [63]: df.plot.hist()
```

```
Out[63]: <AxesSubplot:ylabel='Frequency'>
```



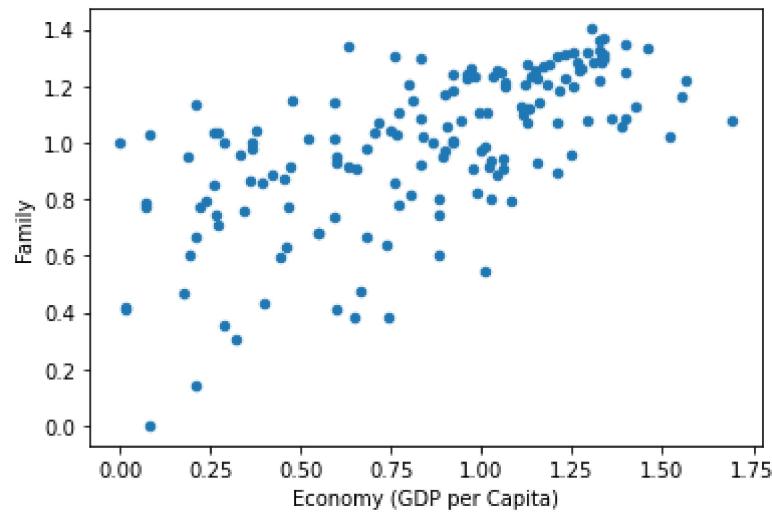
```
In [64]: df.plot.pie(y='Economy (GDP per Capita)')
```

```
Out[64]: <AxesSubplot:ylabel='Economy (GDP per Capita)'>
```



```
In [65]: df.plot.scatter(x='Economy (GDP per Capita)',y='Family')
```

```
Out[65]: <AxesSubplot:xlabel='Economy (GDP per Capita)', ylabel='Family'>
```



```
In [ ]:
```