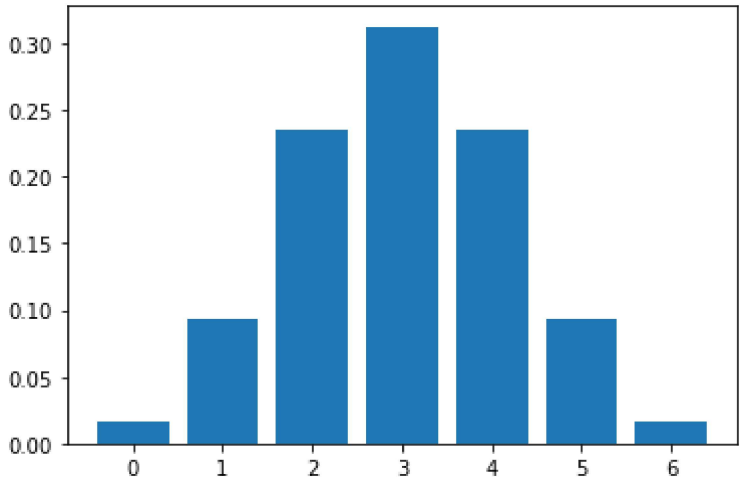
Binomial

In [1]:
```
from scipy.stats import binom
import matplotlib.pyplot as plt
```

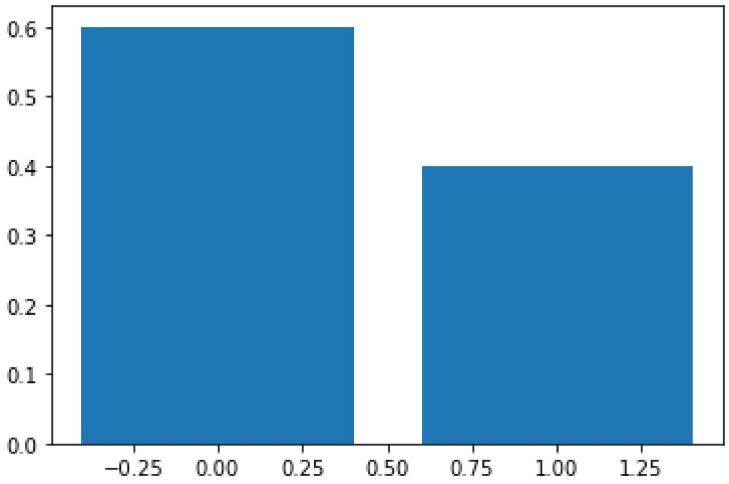
In [2]:
```
n=6
p=0.5
r_values=list(range(n+1))
dist=[binom.pmf(r,n,p) for r in r_values]
plt.bar(r_values,dist)
plt.show()
```



Bernoulli

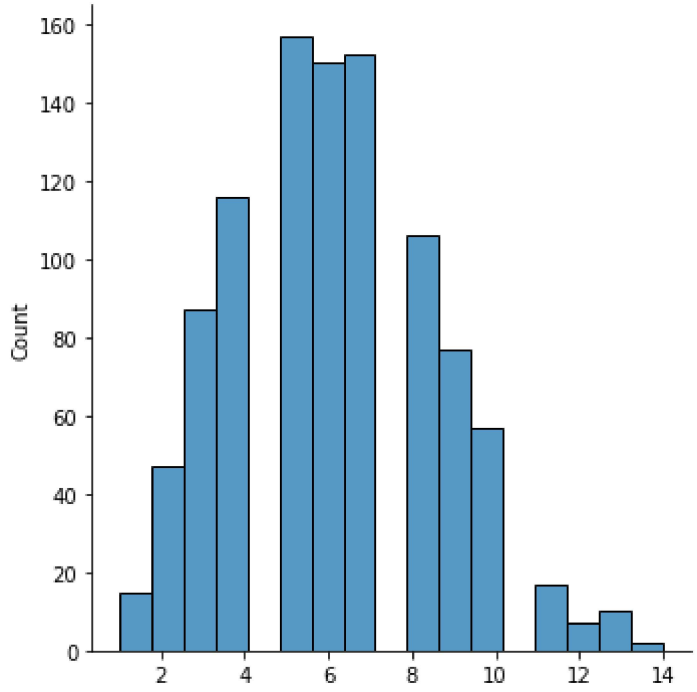
In [3]:
```
import matplotlib.pyplot as plt
from scipy.stats import bernoulli
```

In [4]:
```
bd=bernoulli(0.4)
x=[0,1]
plt.bar(x,bd.pmf(x))
plt.show()
```



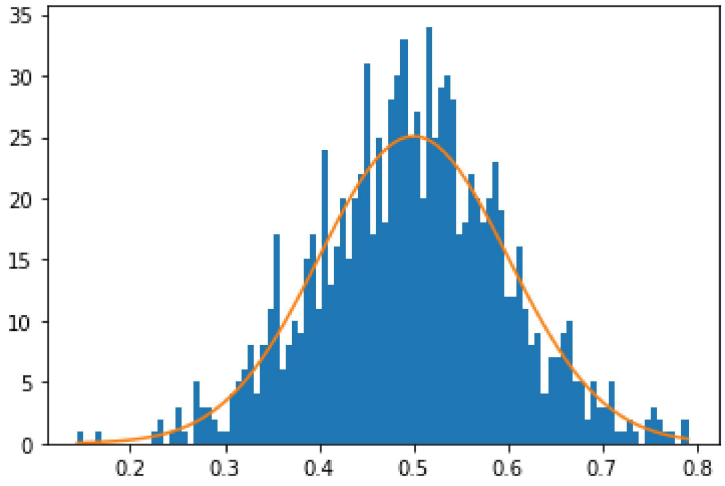
Poisson distribution

In [5]:
```
from numpy import random
import matplotlib.pyplot as plt
import seaborn as sns
sns.displot(random.poisson(lam=6,size=1000))
plt.show()
```



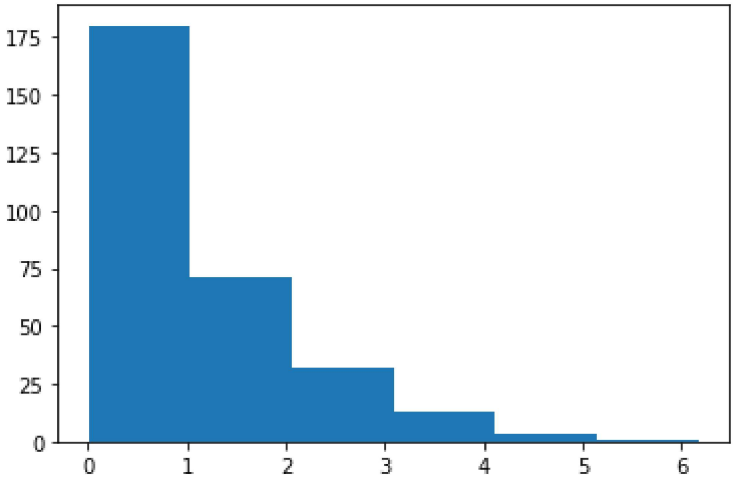
Normal

In [13]:
```
import matplotlib.pyplot as plt
import numpy as np
mu,sigma=0.5,0.1
s=np.random.normal(mu,sigma,1000)
counts,bins,ignored=plt.hist(s,100)
plt.plot(bins, 1/sigma*np.sqrt(2*np.pi)*np.exp(-(bins-mu)**2/(2*sigma**2)))
plt.show()
```



Exponential

In [17]:
```
import numpy as np
import matplotlib.pyplot as plt
exp=np.random.exponential(1,300)
counts,bins,ignored=plt.hist(exp,6)
plt.show()
```



1st Dataset

importing libraries

In [18]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Collecting the data

In [21]:
```
df1=pd.read_csv(r"C:\Users\user\Desktop\Ash\Datasets\4_drug200.csv")
df1
```

Out[21]:

	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

Selecting first 10 rows

In [22]:
```
df1.head(10)
```

Out[22]:

	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

Selecting last 13 rows

In [23]:
```
df1.tail(13)
```

Out[23]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
187	47	M	HIGH	HIGH	10.403	drugA
188	65	M	HIGH	NORMAL	34.997	drugY
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

Statistical data

In [24]:
```
df1.describe()
```

Out[24]:

	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

Shape

In [25]:
```
np.shape(df1)
```

Out[25]:
```
(200, 6)
```

Size

In [26]:
```
np.size(df1)
```

Out[26]:
```
1200
```

In [27]:
```
df1.isna()
```

Out[27]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
195	False	False	False	False	False	False
196	False	False	False	False	False	False
197	False	False	False	False	False	False
198	False	False	False	False	False	False
199	False	False	False	False	False	False

200 rows × 6 columns

In [28]:
```
df1.fillna(value=44)
```

Out[28]:

	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

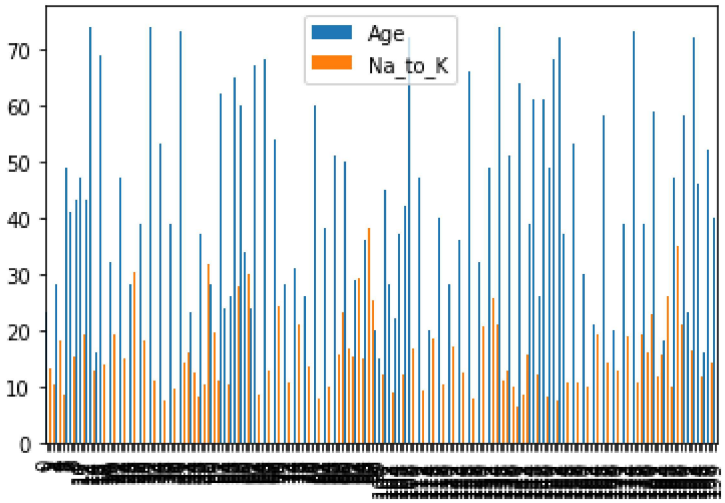
200 rows × 6 columns

Visualization

Bar

In [30]:
```
df1.plot.bar()
```

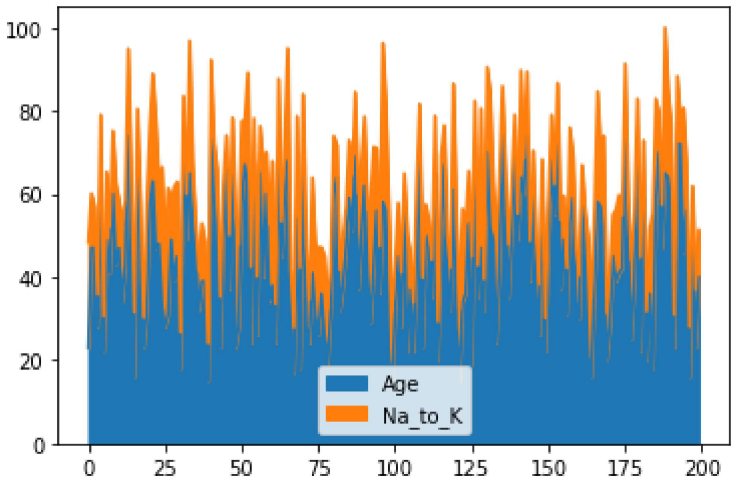
Out[30]: <AxesSubplot:>



Area

In [31]:
```
df1.plot.area()
```

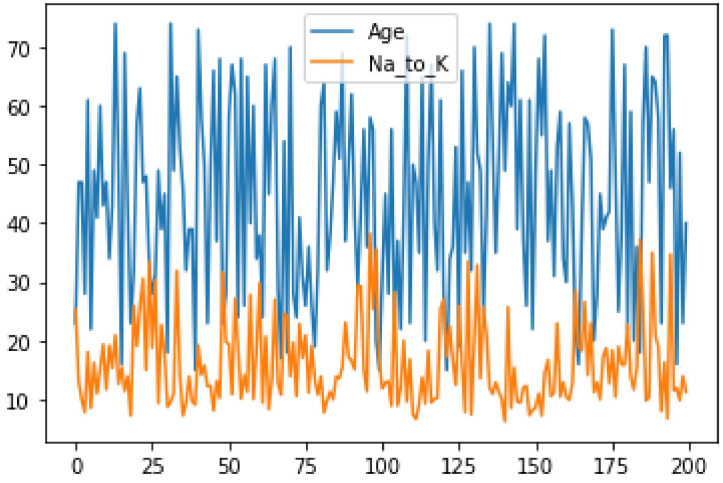
Out[31]: <AxesSubplot:>



Line

In [32]:
```
df1.plot.line()
```

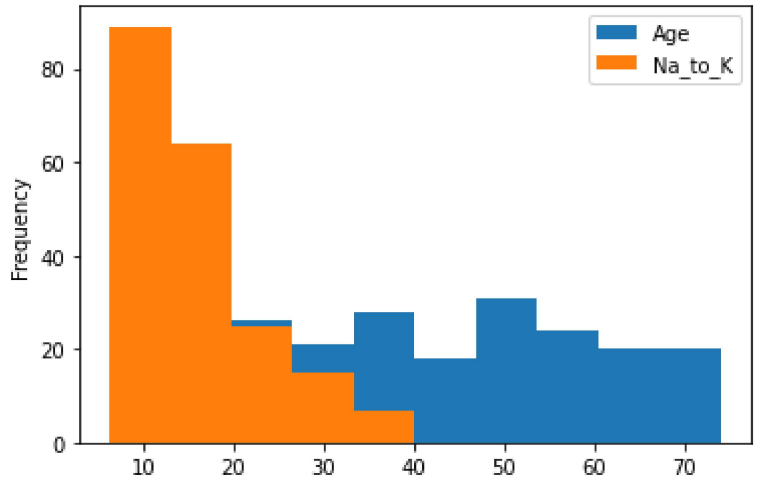
Out[32]: <AxesSubplot:>



Histogram

In [33]:
```
df1.plot.hist()
```

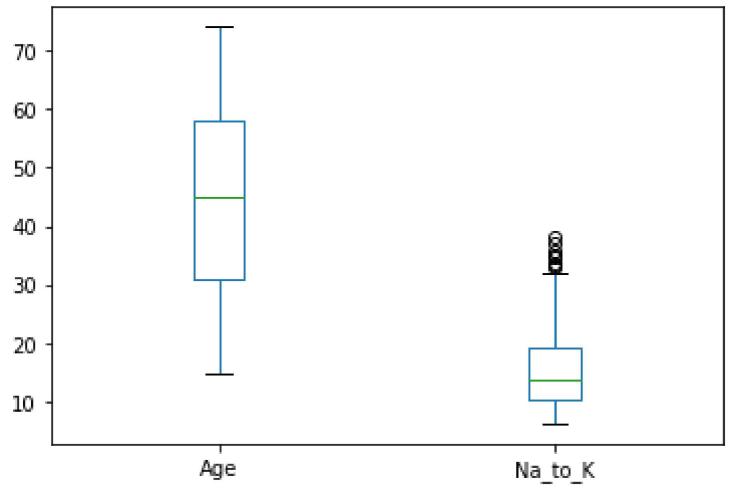
Out[33]: <AxesSubplot:ylabel='Frequency'>



Box

In [34]:
```
df1.plot.box()
```

Out[34]: <AxesSubplot:>



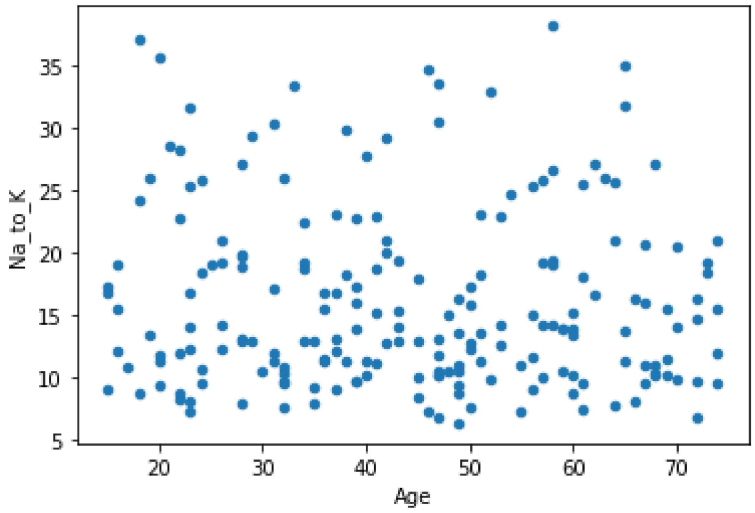
Pie chart

In [37]:
```
df1.plot.pie(y='Age')
```

Scatter chart

In [38]:
```
df1.plot.scatter(x='Age',y='Na_to_K')
```

Out[38]:
```
<AxesSubplot:xlabel='Age', ylabel='Na_to_K'>
```



Dataset 2

Collecting the data

In [40]:
```
df2=pd.read_csv(r"C:\Users\user\Desktop\Ash\Datasets\5_Instagram data.csv")
df2
```

Out[40]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption	
0	3920	2586	1028	619	56	98	9	5	162	35	2	Here are some of the most important data visua...	#finance🔍#money🔍#
1	5394	2727	1838	1174	78	194	7	14	224	48	10	Here are some of the best data science project...	#healthcare🔍#health
2	4021	2085	1188	0	533	41	11	1	131	62	12	Learn how to train a machine learning model an...	#data🔍#datascien
3	4528	2700	621	932	73	172	10	7	213	23	8	Here🔍s how you can write a Python program to d...	#python🔍#pythonpro
4	2518	1704	255	279	37	96	5	4	123	8	0	Plotting annotations while visualizing your da...	#datavisualization🔍
...	...	...	...	...	...	...	...	...	...	...	...	...	
114	13700	5185	3041	5352	77	573	2	38	373	73	80	Here are some of the best data science certifi...	#datascience🔍#d
115	5731	1923	1368	2266	65	135	4	1	148	20	18	Clustering is a machine learning technique use...	#machinelearning🔍

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption	
116	4139	1133	1538	1367	33	36	0	1	92	34	10	Clustering music genres is a task of grouping ...	#machinelearning[illegible]
117	32695	11815	3147	17414	170	1095	2	75	549	148	214	Here are some of the best data science certifi...	#datascience[illegible]#d
118	36919	13473	4176	16444	2547	653	5	26	443	611	228	175 Python Projects with Source Code solved an...	#python[illegible]#pythonpro

119 rows × 13 columns

Selecting first 13 rows

In [41]:
```
df2.head(13)
```

Out[41]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption	
0	3920	2586	1028	619	56	98	9	5	162	35	2	Here are some of the most important data visua...	#finance#money
1	5394	2727	1838	1174	78	194	7	14	224	48	10	Here are some of the best data science project...	#healthcare#heal
2	4021	2085	1188	0	533	41	11	1	131	62	12	Learn how to train a machine learning model an...	#data#datasci
3	4528	2700	621	932	73	172	10	7	213	23	8	Here's how you can write a Python program to d...	#python#pythonp
4	2518	1704	255	279	37	96	5	4	123	8	0	Plotting annotations while visualizing your da...	#datavisualizatio
5	3884	2046	1214	329	43	74	7	10	144	9	2	Here are some of the most important soft skill...	#data#datasci
6	2621	1543	599	333	25	22	5	1	76	26	0	Learn how to analyze a candlestick chart as a ...	#stockmarket#in
7	3541	2071	628	500	60	135	4	9	124	12	6	Here are some of the best books that you can f...	#python#pythonp
8	3749	2384	857	248	49	155	6	8	159	36	4	Here are some of the best data analysis projec...	#dataanalytics#

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption	
9	4115	2609	1104	178	46	122	6	3	191	31	6	Here are two best ways to count the number of ...	#python#pythonp
10	2218	1597	411	162	15	28	6	3	81	29	4	Learn the implementation of AlexNet Convolutio...	#neuralnetw
11	3234	2414	476	185	75	122	8	14	151	15	0	Here's how to get the live stock price data of...	#python#pythonp
12	4344	2168	1274	673	40	119	7	11	162	8	2	Here are some of the most important Python lib...	#data#datasci

Selecting last 8 rows

In [42]:
```
df2.tail(8)
```

Out[42]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption	
111	4842	1658	694	2036	310	55	6	4	86	46	30	Learn how to create an interactive language tr...	#python#pythonpr
112	11149	4439	747	5762	53	273	4	13	210	61	58	Python is one of the best programming language...	#python#pythonpr
113	10206	2371	1624	6000	117	182	10	17	172	237	100	Practice these 90+ Data Science Projects For B...	#datascience
114	13700	5185	3041	5352	77	573	2	38	373	73	80	Here are some of the best data science certifi...	#datascience
115	5731	1923	1368	2266	65	135	4	1	148	20	18	Clustering is a machine learning technique use...	#machinelearnin
116	4139	1133	1538	1367	33	36	0	1	92	34	10	Clustering music genres is a task of grouping ...	#machinelearnin
117	32695	11815	3147	17414	170	1095	2	75	549	148	214	Here are some of the best data science certifi...	#datascience
118	36919	13473	4176	16444	2547	653	5	26	443	611	228	175 Python Projects with Source Code solved an...	#python#pythonp

statistical data

In [43]:
```
df2.describe()
```

Out[43]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profi Visi
count	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.0000(
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.663866	9.361345	173.781513	50.6218
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.544576	10.089205	82.378947	87.0884(
min	1941.000000	1133.000000	116.000000	0.000000	9.000000	22.000000	0.000000	0.000000	72.000000	4.0000(
25%	3467.000000	1945.000000	726.000000	157.500000	38.000000	65.000000	4.000000	3.000000	121.500000	15.0000(
50%	4289.000000	2207.000000	1278.000000	326.000000	74.000000	109.000000	6.000000	6.000000	151.000000	23.0000(
75%	6138.000000	2602.500000	2363.500000	689.500000	196.000000	169.000000	8.000000	13.500000	204.000000	42.0000(
max	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	1095.000000	19.000000	75.000000	549.000000	611.0000(

Shape

In [44]:
```
np.shape(df2)
```

Out[44]:
```
(119, 13)
```

Size

In [45]:
```
np.size(df2)
```

Out[45]:
```
1547
```

Finding missing values

In [46]:
```
df2.isna()
```

Out[46]:

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

119 rows × 13 columns

Deleting columns having empty cells

In [47]:
```
df2.dropna()
```

Out[47]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption	
0	3920	2586	1028	619	56	98	9	5	162	35	2	Here are some of the most important data visua...	#finance#money#
1	5394	2727	1838	1174	78	194	7	14	224	48	10	Here are some of the best data science project...	#healthcare#health
2	4021	2085	1188	0	533	41	11	1	131	62	12	Learn how to train a machine learning model an...	#data#datascien
3	4528	2700	621	932	73	172	10	7	213	23	8	Here's how you can write a Python program to d...	#python#pythonpro
4	2518	1704	255	279	37	96	5	4	123	8	0	Plotting annotations while visualizing your da...	#datavisualization
...	...	...	...	...	...	...	...	...	...	...	...	...	
114	13700	5185	3041	5352	77	573	2	38	373	73	80	Here are some of the best data science certifi...	#datascience#d
115	5731	1923	1368	2266	65	135	4	1	148	20	18	Clustering is a machine learning technique use...	#machinelearning

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption	
116	4139	1133	1538	1367	33	36	0	1	92	34	10	Clustering music genres is a task of grouping ...	#machinelearning[illegible]
117	32695	11815	3147	17414	170	1095	2	75	549	148	214	Here are some of the best data science certifi...	#datascience[illegible]#d
118	36919	13473	4176	16444	2547	653	5	26	443	611	228	175 Python Projects with Source Code solved an...	#python[illegible]#pythonpro

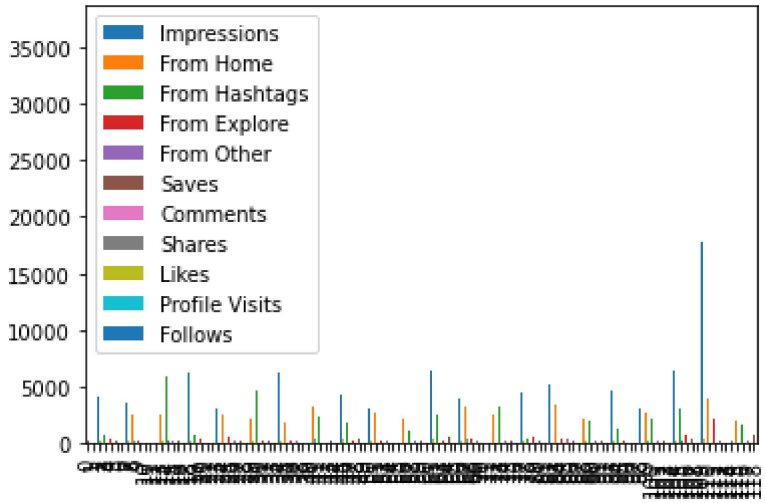
119 rows × 13 columns

visualization

Bar graph

In [48]:
```
df2.plot.bar()
```

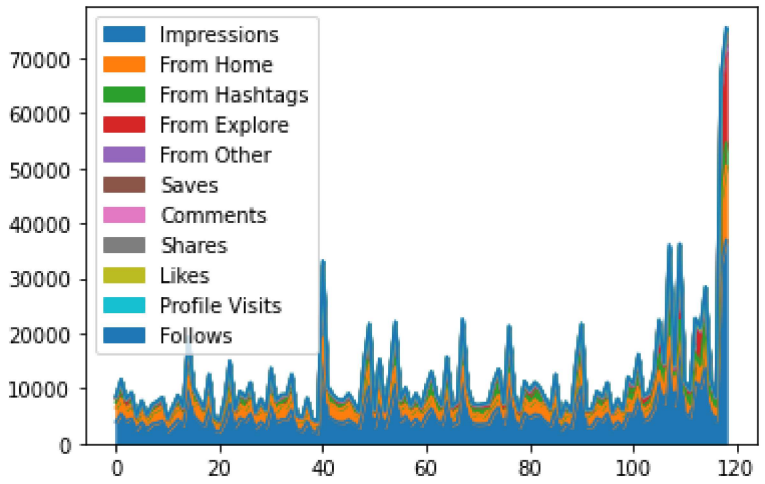
Out[48]: <AxesSubplot:>



Area graph

In [49]:
```
df2.plot.area()
```

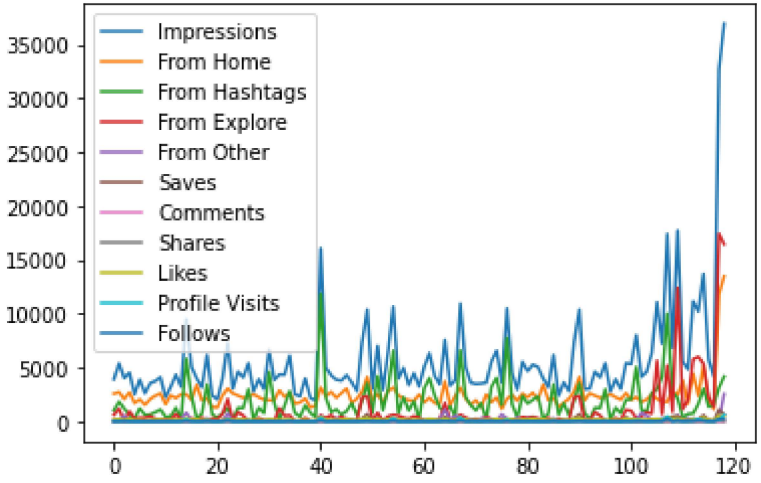
Out[49]: <AxesSubplot:>



Line graph

In [50]:
```
df2.plot.line()
```

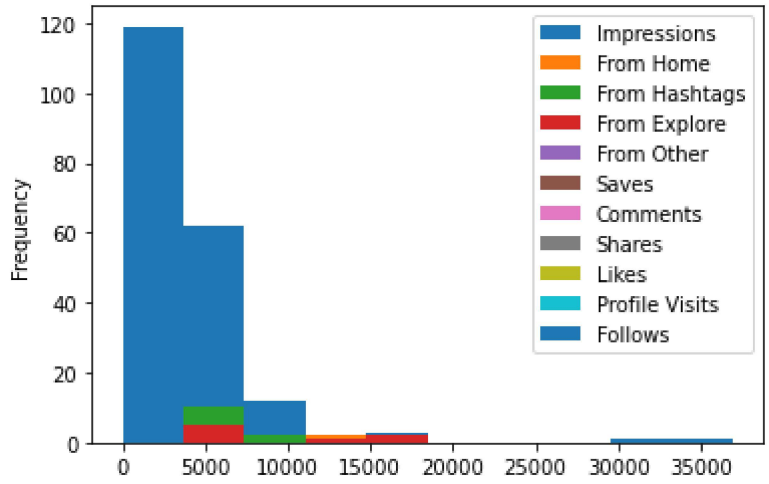
Out[50]: <AxesSubplot:>



Histogram

In [51]:
```
df2.plot.hist()
```

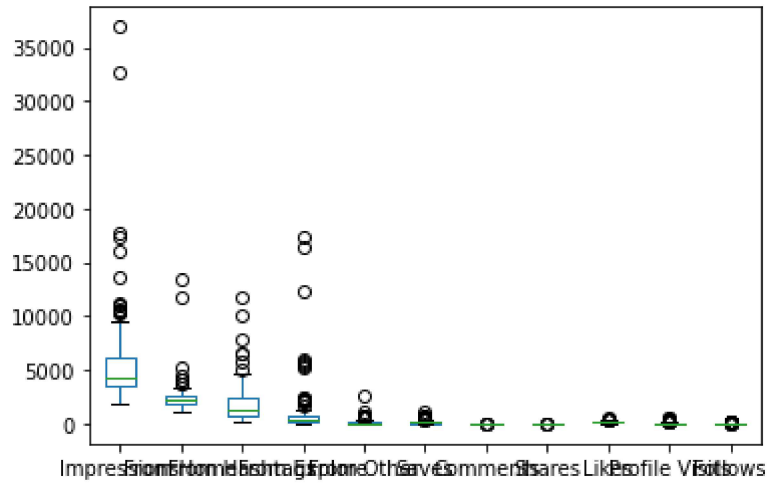
Out[51]: <AxesSubplot:ylabel='Frequency'>



Box chart

In [53]:
```
df2.plot.box()
```

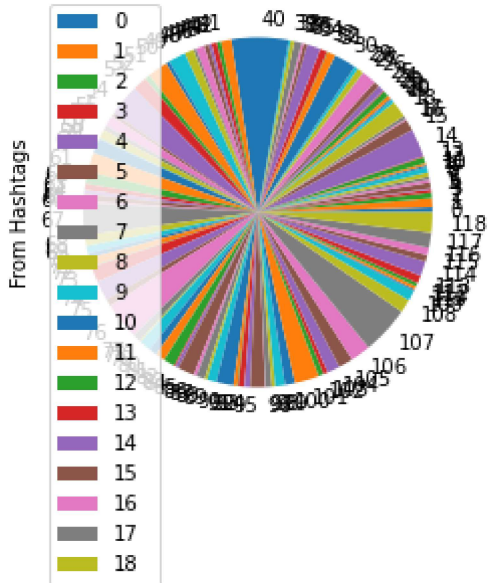
Out[53]: <AxesSubplot:>



Pie chart

In [54]:
```
df2.plot.pie(y='From Hashtags')
```

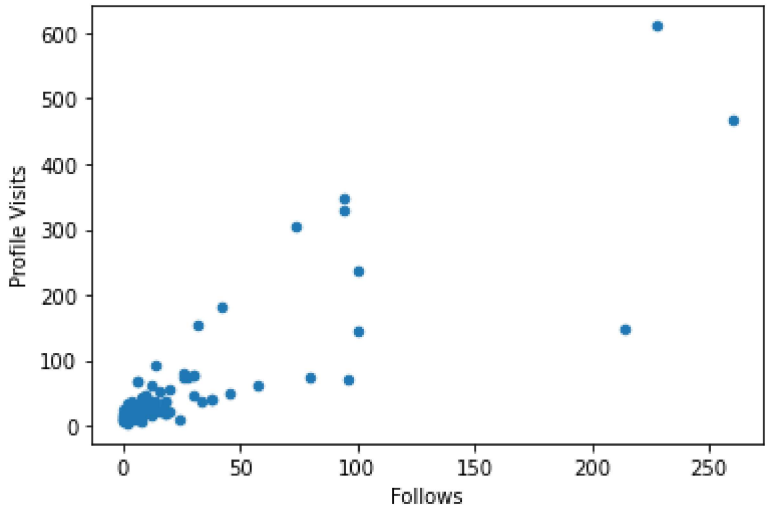
Out[54]:
```
<AxesSubplot:ylabel='From Hashtags'>
```



Scatter plot

In [55]:
```
df2.plot.scatter(x='Follows',y='Profile Visits')
```

Out[55]:
```
<AxesSubplot:xlabel='Follows', ylabel='Profile Visits'>
```



Dataset 3

Importing the data

In [57]:
```
df3=pd.read_csv(r"C:\Users\user\Desktop\Ash\Datasets\2015.csv")
df3
```

Out[57]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystop Residu
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.517
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.702
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.492
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.465
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.451
...	...	...	...	...	...	...	...	...	...	...	...	
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.670
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.633
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.328
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.833
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.567

158 rows × 12 columns

Selecting top 12 rows

In [58]:
```
df3.head(12)
```

Out[58]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dysto Resid
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45
5	Finland	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	0.64169	0.41372	0.23351	2.61
6	Netherlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	0.61576	0.31814	0.47610	2.46
7	Sweden	Western Europe	8	7.364	0.03157	1.33171	1.28907	0.91087	0.65980	0.43844	0.36262	2.37
8	New Zealand	Australia and New Zealand	9	7.286	0.03371	1.25018	1.31967	0.90837	0.63938	0.42922	0.47501	2.26
9	Australia	Australia and New Zealand	10	7.284	0.04083	1.33358	1.30923	0.93156	0.65124	0.35637	0.43562	2.26
10	Israel	Middle East and Northern Africa	11	7.278	0.03470	1.22857	1.22393	0.91387	0.41319	0.07785	0.33172	3.08
11	Costa Rica	Latin America and Caribbean	12	7.226	0.04454	0.95578	1.23788	0.86027	0.63376	0.10583	0.25497	3.17

Selecting last 9 rows

In [59]:
```
df3.tail(9)
```

Out[59]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dysto Resid
149	Guinea	Sub-Saharan Africa	150	3.656	0.03590	0.17417	0.46475	0.24009	0.37725	0.12139	0.28657	1.99
150	Ivory Coast	Sub-Saharan Africa	151	3.655	0.05141	0.46534	0.77115	0.15185	0.46866	0.17922	0.20165	1.41
151	Burkina Faso	Sub-Saharan Africa	152	3.587	0.04324	0.25812	0.85188	0.27125	0.39493	0.12832	0.21747	1.46
152	Afghanistan	Southern Asia	153	3.575	0.03084	0.31982	0.30285	0.30335	0.23414	0.09719	0.36510	1.95
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.67
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.63
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.32
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.83
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.56

Statistical data

In [60]:
```
df3.describe()
```

Out[60]:

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
count	158.000000	158.000000	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	0.143422	0.237296	2.098977
std	45.754363	1.145010	0.017146	0.403121	0.272369	0.247078	0.150693	0.120034	0.126685	0.553550
min	1.000000	2.839000	0.018480	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.328580
25%	40.250000	4.526000	0.037268	0.545808	0.856823	0.439185	0.328330	0.061675	0.150553	1.759410
50%	79.500000	5.232500	0.043940	0.910245	1.029510	0.696705	0.435515	0.107220	0.216130	2.095415
75%	118.750000	6.243750	0.052300	1.158448	1.214405	0.811013	0.549092	0.180255	0.309883	2.462415
max	158.000000	7.587000	0.136930	1.690420	1.402230	1.025250	0.669730	0.551910	0.795880	3.602140

Shape

In [61]:
```
np.shape(df3)
```

Out[61]:
```
(158, 12)
```

Size

In [62]:
```
np.size(df3)
```

Out[62]:
```
1896
```

Finding missing values

In [63]:
```
df3.isna()
```

Out[63]:

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

158 rows × 12 columns

Filling missing values with 54

In [64]:
```
df3.fillna(value=54)
```

Out[64]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystop Residu
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.517
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.702
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.492
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.465
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.451
...	...	...	...	...	...	...	...	...	...	...	...	
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	0.59201	0.55191	0.22628	0.670
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	0.48450	0.08010	0.18260	1.633
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	0.15684	0.18906	0.47179	0.328
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727	1.833
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	0.36453	0.10731	0.16681	1.567

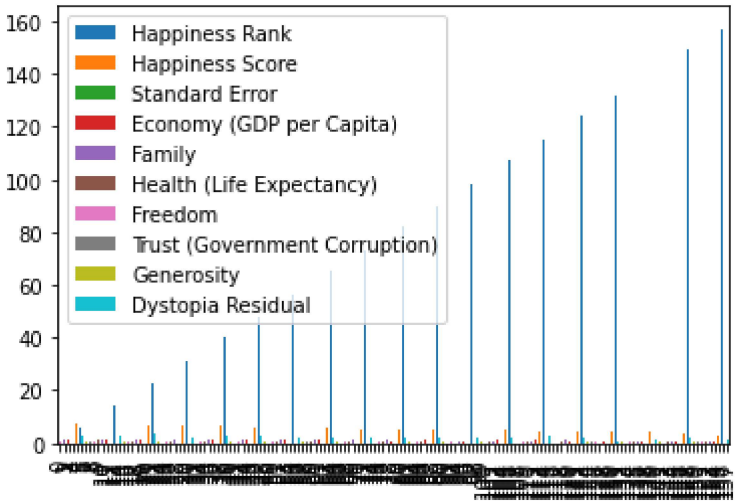
158 rows × 12 columns

Visualization

Bar chart

In [65]:
```
df3.plot.bar()
```

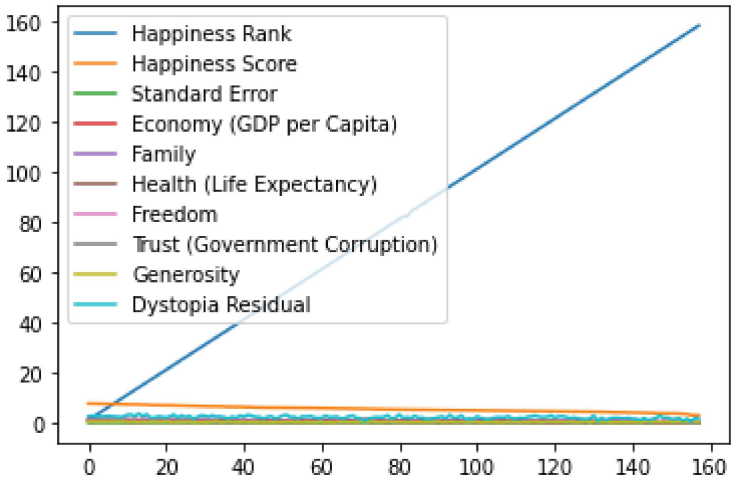
Out[65]: <AxesSubplot:>



Line chart

In [66]:
```
df3.plot.line()
```

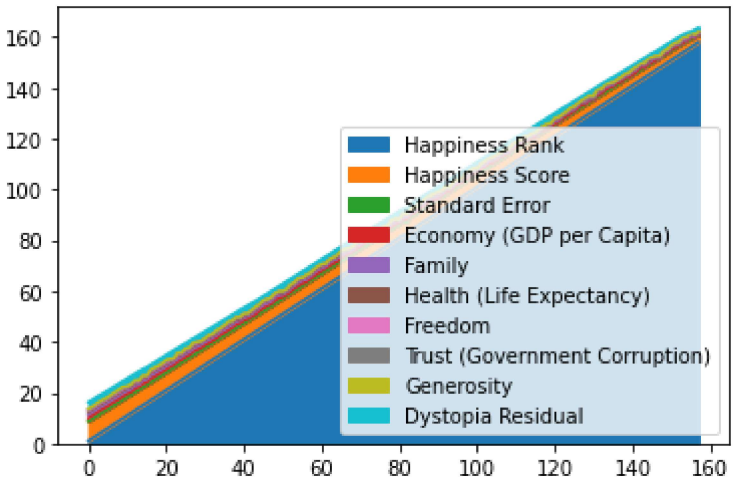
Out[66]: <AxesSubplot:>



Area Chart

In [67]:
```
df3.plot.area()
```

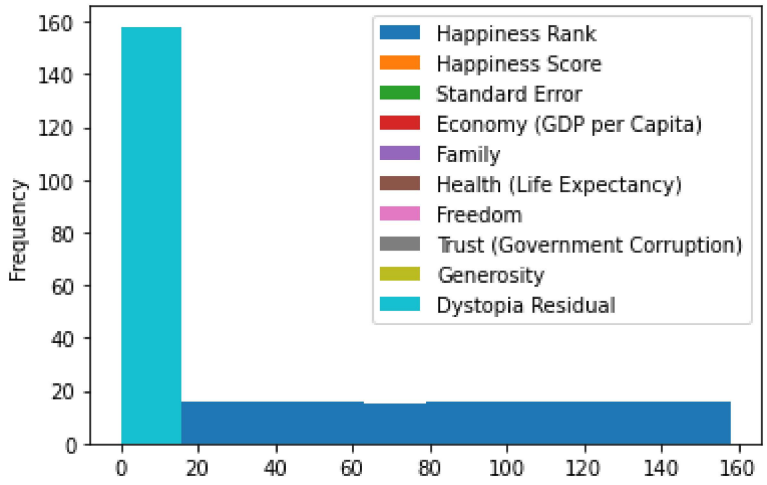
Out[67]: <AxesSubplot:>



Histogram

In [68]:
```
df3.plot.hist()
```

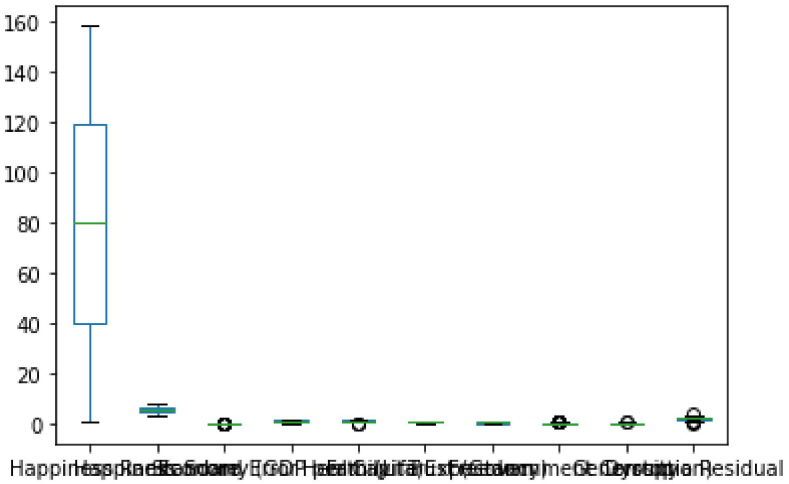
Out[68]:
```
<AxesSubplot:ylabel='Frequency'>
```



Box chart

In [69]:
```
df3.plot.box()
```

Out[69]: <AxesSubplot:>



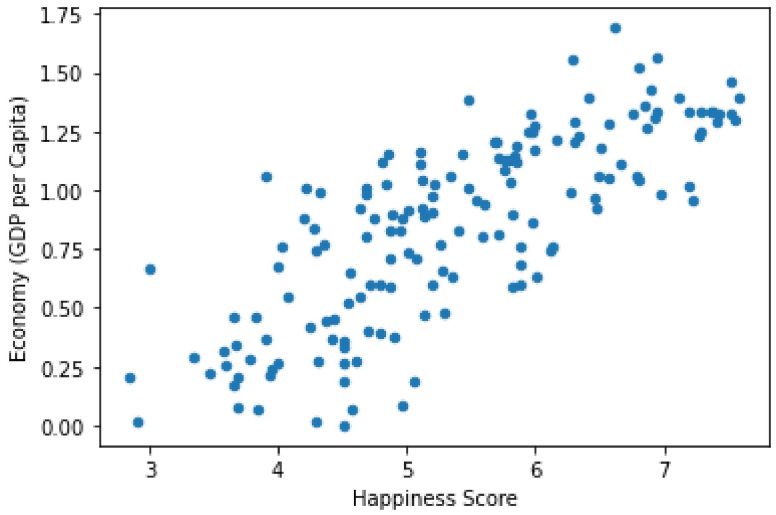
Pie chart

In [74]:
```
df3.plot.pie(y='Health (Life Expectancy)')
```

Scatter plot

In [75]:
```
df3.plot.scatter(x='Happiness Score',y='Economy (GDP per Capita)')
```

Out[75]:
```
<AxesSubplot:xlabel='Happiness Score', ylabel='Economy (GDP per Capita)'>
```



In [ ]: