In [1]:

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

In [2]:

```
b=pd.read_csv(r"E:\Python Data Science\Documents\2015.csv")
```

```
To print 1st Five Rows
```

In [3]:

```
b.head(8)
```

Out[3]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	F
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	
5	Finland	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	
6	Netherlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	
7	Sweden	Western Europe	8	7.364	0.03157	1.33171	1.28907	0.91087	

```
To print last 5 rows
```

In [4]:

```
b.tail()
```

Out[4]:

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Fr
153	Rwanda	Sub-Saharan Africa	154	3.465	0.03464	0.22208	0.77370	0.42864	(
154	Benin	Sub-Saharan Africa	155	3.340	0.03656	0.28665	0.35386	0.31910	(
155	Syria	Middle East and Northern Africa	156	3.006	0.05015	0.66320	0.47489	0.72193	(
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	(
157	Togo	Sub-Saharan Africa	158	2.839	0.06727	0.20868	0.13995	0.28443	(

To find shape

In [5]:

```
b.shape
```

Out[5]:

```
(158, 12)
```

To find size

In [6]:

```
b.size
```

Out[6]:

```
1896
```

To describe

In [7]:

```
b.describe()
```

Out[7]:

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
count	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

To check the null values

In [8]:

```
b.isna()
```

Out[8]:

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

158 rows × 12 columns

To fill the null value

In [9]:

```
b.fillna(value=5)
```

Out[9]:

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

158 rows × 12 columns

```
To drop the null valued rows
```

In [10]:

```
b.dropna()
```

Out[10]:

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

158 rows × 12 columns

In [11]:

```
conda install matplotlib
```

```
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done

# All requested packages already installed.


Note: you may need to restart the kernel to use updated packages.



==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 23.5.2

Please update conda by running

    $ conda update -n base -c defaults conda
```

In [12]:

```
import matplotlib.pyplot as pp
```

In [15]:

```
c=b[['Economy (GDP per Capita)','Family']]
c
```

Out[15]:

	Economy (GDP per Capita)	Family
0	1.39651	1.34951
1	1.30232	1.40223
2	1.32548	1.36058
3	1.45900	1.33095
4	1.32629	1.32261
...	...	...
153	0.22208	0.77370
154	0.28665	0.35386
155	0.66320	0.47489
156	0.01530	0.41587
157	0.20868	0.13995

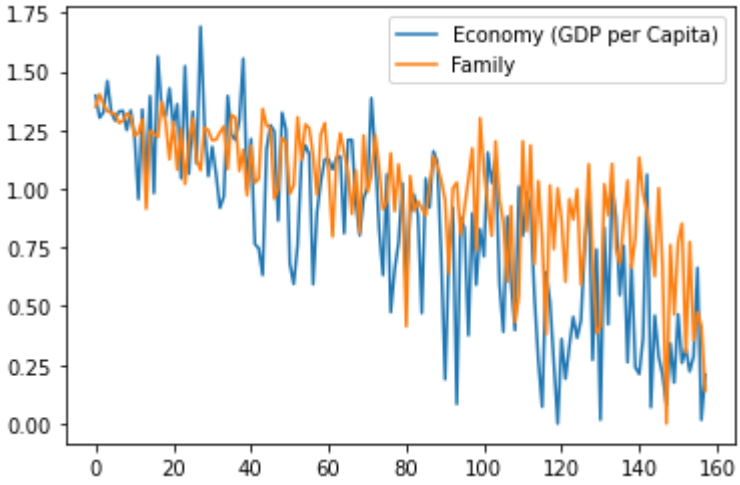
158 rows × 2 columns

In [16]:

```
c.plot.line()
```

Out[16]:

```
<AxesSubplot:>
```

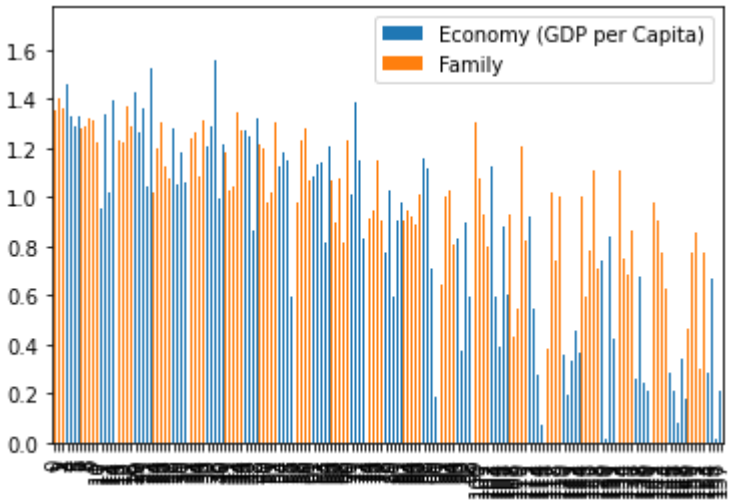


In [17]:

```
c.plot.bar()
```

Out[17]:

```
<AxesSubplot:>
```

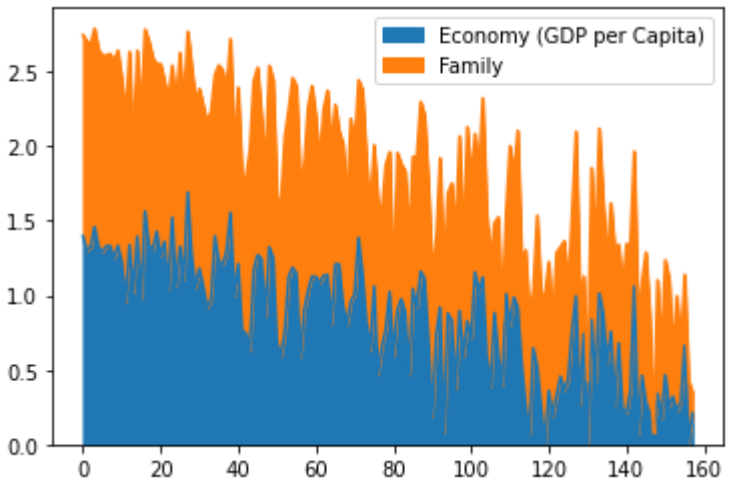


In [18]:

```
c.plot.area()
```

Out[18]:

<AxesSubplot:>

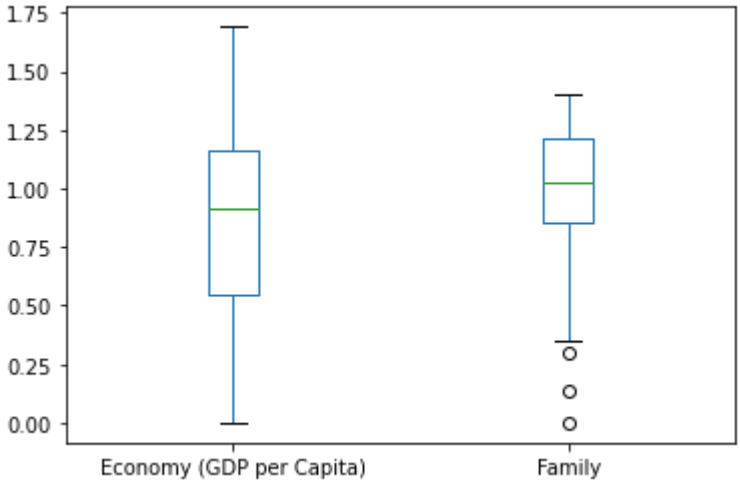


In [19]:

```
c.plot.box()
```

Out[19]:

<AxesSubplot:>

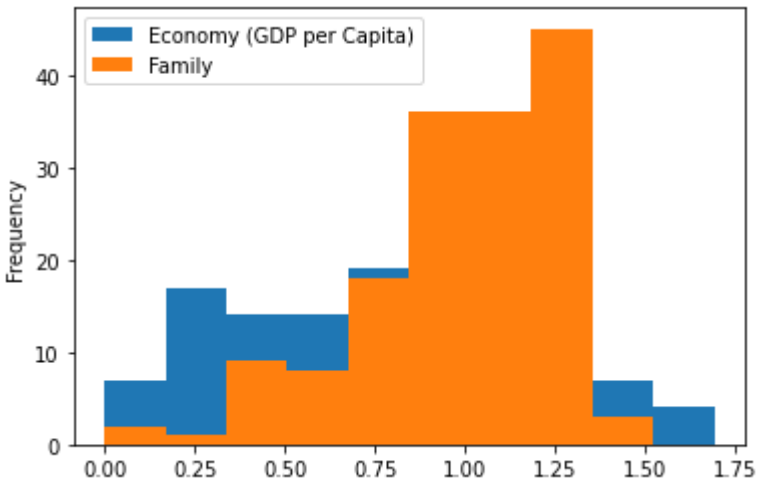


In [20]:

```
c.plot.hist()
```

Out[20]:

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

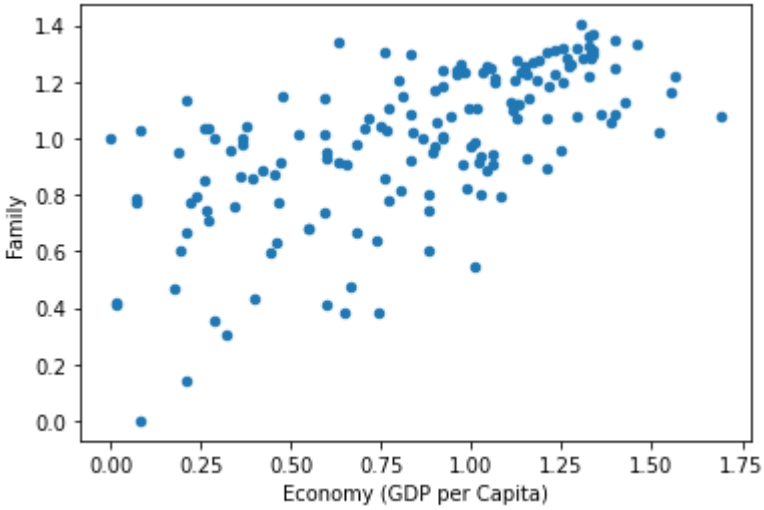


In [23]:

```
c.plot.scatter(x='Economy (GDP per Capita)',y='Family')
```

Out[23]:

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

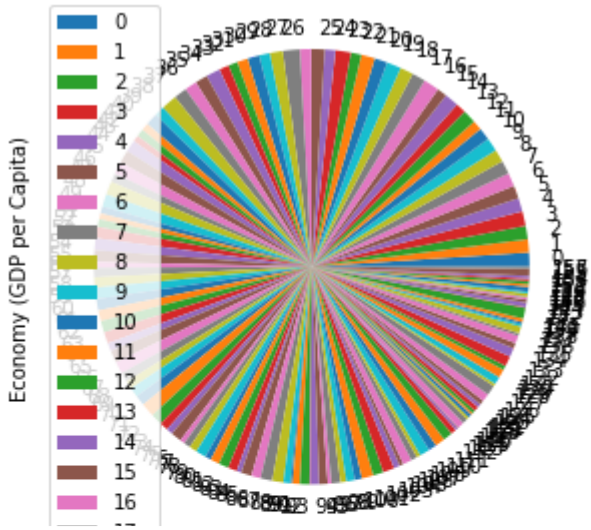


In [24]:

```
c.plot.pie(y='Economy (GDP per Capita)',figsize=(5,5))
```

Out[24]:

```
<AxesSubplot:ylabel='Economy (GDP per Capita)'>
```



In [ ]:

```

```