```
In [2]: import numpy as np

In [3]: one = np.array([1, 8, 9])

In [4]: one

Out[4]: array([1, 8, 9])

In [5]: one.ndim

Out[5]: 1

In [6]: two = np.array([[1, 2, 3]])

In [7]: two.ndim

Out[7]: 2

In [8]: five = np.array([[[[[1, 2, 3, 4, 5]]]]])

In [9]: five.ndim

Out[9]: 5

In [10]: two.shape

Out[10]: (1, 3)

In [11]: two.size

Out[11]: 3

In [12]: two.dtype

Out[12]: dtype('int32')

In [13]: two.itemsize

Out[13]: 4

In [14]: one_arr = np.ones((3,3))
         one_arr

Out[14]: array([[1., 1., 1.],
                [1., 1., 1.],
                [1., 1., 1.]])

In [15]: zero_arr = np.zeros((3,3))
         zero_arr

Out[15]: array([[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]])

In [16]: random = np.random.random((3,3))
         random

Out[16]: array([[0.97099809, 0.85230672, 0.28090301],
                [0.72133309, 0.63134075, 0.65347414],
                [0.22593167, 0.68290811, 0.3551073 ]])
```

In [17]:

```
identity = np.identity(3)
identity
```

Out[17]:

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

In [18]:

```
arrange = np.arange(5, 11)
arrange
```

Out[18]:

```
array([ 5,  6,  7,  8,  9, 10])
```

In [19]:

```
linspace = np.linspace(2, 10, 2)
linspace
```

Out[19]:

```
array([ 2., 10.])
```

In [20]:

```
linspace1 = np.linspace(2, 10, 20)
linspace1
```

Out[20]:

```
array([ 2.        ,  2.42105263,  2.84210526,  3.26315789,  3.68421053,
        4.10526316,  4.52631579,  4.94736842,  5.36842105,  5.78947368,
        6.21052632,  6.63157895,  7.05263158,  7.47368421,  7.89473684,
        8.31578947,  8.73684211,  9.15789474,  9.57894737, 10.        ])
```

In [21]:

```
my_list = [0, 1, 2, 3, 4, 5]
my_list
```

Out[21]:

```
[0, 1, 2, 3, 4, 5]
```

In [22]:

```
arr = np.array(my_list)
arr
```

Out[22]:

```
array([0, 1, 2, 3, 4, 5])
```

In [23]:

```
my_tuple = {1, 2, 3, 4, 5, 6}
my_arr = np.array(my_tuple)
my_arr
```

Out[23]:

```
array({1, 2, 3, 4, 5, 6}, dtype=object)
```

In [24]:

```
# array = [0, 1, 2, 3, 4, 5]
arr[::2]
```

Out[24]:

```
array([0, 2, 4])
```

In [25]:

```
arr[::-1]
```

Out[25]:

```
array([5, 4, 3, 2, 1, 0])
```

In [26]:

```
kernel = np.array([[1, 2, 3],[5, 6, 7]])
kernel
```

Out[26]:

```
array([[1, 2, 3],
       [5, 6, 7]])
```

In [27]:

```
kernel.flatten()
```

Out[27]:

```
array([1, 2, 3, 5, 6, 7])
```

In [28]:
```
array1 = np.array([[[1, 2, 3, 4, 5, 6, 7], [8, 9, 10, 11, 12, 13, 14]]])
array1
```

Out[28]:
```
array([[[ 1,  2,  3,  4,  5,  6,  7],
        [ 8,  9, 10, 11, 12, 13, 14]]])
```

In [29]:
```
array1.flatten()
```

Out[29]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

In [30]:
```
kernel.size
```

Out[30]:
```
6
```

In [31]:
```
print(kernel.resize(3,2))
```

```
None
```

In [32]:
```
my_arr = np.array([[0,1],[2,3]])
my_arr
```

Out[32]:
```
array([[0, 1],
       [2, 3]])
```

In [33]:
```
my_arr = np.resize(1,6)
my_arr
```

Out[33]:
```
array([1, 1, 1, 1, 1, 1])
```

In [34]:
```
np.resize(my_arr,(2,3))
```

Out[34]:
```
array([[1, 1, 1],
       [1, 1, 1]])
```

In [35]:
```
import pandas as pd
```

In [36]:
```
my_dict = {"Name" : ["Siddesh" , "Sahil", "Shreyash", "Vedant"],
           "Roll.no" : ["39010", "39019", "39014", "39007"],
           "Age" : ["20", "21", "20", "22"],
           "Attendance" : ["100", "96", "93", "90"]
          }
my_dict
```

Out[36]:
```
{'Name': ['Siddesh', 'Sahil', 'Shreyash', 'Vedant'],
 'Roll.no': ['39010', '39019', '39014', '39007'],
 'Age': ['20', '21', '20', '22'],
 'Attendance': ['100', '96', '93', '90']}
```

In [37]:
```
df = pd.DataFrame(my_dict)
df
```

Out[37]:

	Name	Roll.no	Age	Attendance
0	Siddesh	39010	20	100
1	Sahil	39019	21	96
2	Shreyash	39014	20	93
3	Vedant	39007	22	90

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

Out[38]:
```
<bound method NDFrame.describe of        Name Roll.no Age Attendance
0   Siddesh   39010  20        100
1     Sahil   39019  21         96
2  Shreyash   39014  20         93
3    Vedant   39007  22         90>
```

In [39]:
```
df.describe()
```

Out[39]:

	Name	Roll.no	Age	Attendance
count	4	4	4	4
unique	4	4	3	4
top	Siddesh	39010	20	100
freq	1	1	2	1

In [40]:
```
import numpy as np
import pandas as pd
```

In [41]:
```
data = {
    "Energy Source": ["Solar" , "Wind" , "Hydropower" , "Geothermal" , "Biomass" , "Nuclear"]
    "Energy Consumption":[1200 , np.nan , 2900 , np.nan , 2500 , 3200],
    "Cost Millions ":[200 , 400 , np.nan , 150 , 250 , np.nan]
}
```

In [42]:
```
energy_df = pd.DataFrame(data)
energy_df
```

Out[42]:

	Energy Source	Energy Consumption	Cost Millions
0	Solar	1200.0	200.0
1	Wind	NaN	400.0
2	Hydropower	2900.0	NaN
3	Geothermal	NaN	150.0
4	Biomass	2500.0	250.0
5	Nuclear	3200.0	NaN

In [43]:
```
energy_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Energy Source       6 non-null      object 
 1   Energy Consumption  4 non-null      float64
 2   Cost Millions       4 non-null      float64
dtypes: float64(2), object(1)
memory usage: 276.0+ bytes
```

In [44]:
```
energy_df.isna().sum()
```

Out[44]:
```
Energy Source         0
Energy Consumption    2
Cost Millions         2
dtype: int64
```

In [45]:
```
energy_df.describe()
```

Out[45]:

	Energy Consumption	Cost Millions
count	4.000000	4.000000
mean	2450.000000	250.000000
std	881.286938	108.012345
min	1200.000000	150.000000
25%	2175.000000	187.500000
50%	2700.000000	225.000000
75%	2975.000000	287.500000
max	3200.000000	400.000000

In [65]:
```
import seaborn as sea
```

In [66]:
```
cleaned_df = energy_df.dropna()
```

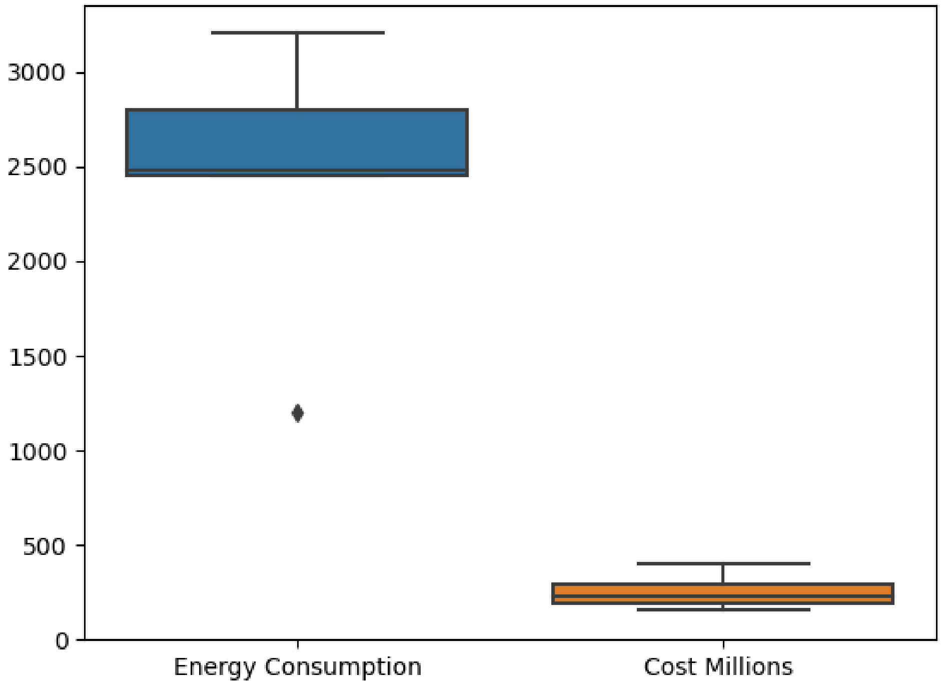
In [67]:
```
cleaned_df
```

Out[67]:

	Energy Source	Energy Consumption	Cost Millions
0	Solar	1200.0	200.0
1	Wind	2450.0	400.0
3	Geothermal	2450.0	150.0
4	Biomass	2500.0	250.0

In [68]:
```
sea.boxplot(energy_df)
```

Out[68]:
```
<Axes: >
```



In [69]:
```
energy_df.columns
```

Out[69]:
```
Index(['Energy Source', 'Energy Consumption', 'Cost Millions '], dtype='object')
```

In [70]:
```
energy_df['Energy Consumption'].fillna(energy_df['Energy Consumption'].mean(), inplace = True
```

In [71]:
```
energy_df
```

Out[71]:

	Energy Source	Energy Consumption	Cost Millions
0	Solar	1200.0	200.0
1	Wind	2450.0	400.0
2	Hydropower	2900.0	NaN
3	Geothermal	2450.0	150.0
4	Biomass	2500.0	250.0
5	Nuclear	3200.0	NaN

In [73]:
```
energy_df["Cost Millions "].fillna(energy_df["Cost Millions "].mean(), inplace = True )

energy_df
```

Out[73]:

	Energy Source	Energy Consumption	Cost Millions
0	Solar	1200.0	200.0
1	Wind	2450.0	400.0
2	Hydropower	2900.0	250.0
3	Geothermal	2450.0	150.0
4	Biomass	2500.0	250.0
5	Nuclear	3200.0	250.0

In [74]:
```
energy_df.isnull().sum()
```

Out[74]:
```
Energy Source         0
Energy Consumption    0
Cost Millions         0
dtype: int64
```

In [75]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [104]:
```
scaler = MinMaxScaler()
```

In [105]:
```
energy_df.columns
```

Out[105]:
```
Index(['Energy Source', 'Energy Consumption', 'Cost Millions ',
       'Cost Millions'],
      dtype='object')
```

In [ ]:
```

```