In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv("C:/Users/Acer/Downloads/Employee (1).csv")
df
```

Out[2]:

	Company	Age	Salary	Place	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
143	TCS	33.0	9024.0	Calcutta	India	1
144	Infosys	22.0	8787.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

148 rows × 6 columns

In [3]:
```
len(df)
```

Out[3]:
```
148
```

In [4]:
```
df.columns
```

Out[4]:
```
Index(['Company', 'Age', 'Salary', 'Place', 'Country', 'Gender'], dtype='object')
```

DATA EXPLORATION

In [5]:
```
# Explore the data, list down the unique values in each feature and find its length
```

In [6]:
```
for i in df.columns:
    x=df[i].unique()

    print(i,': columns',x,':',len(x),'\n')
```

```
Company : columns ['TCS' 'Infosys' 'CTS' nan 'Tata Consultancy Services' 'Congnizant'
 'Infosys Pvt Lmt'] : 7

Age : columns [20. 30. 35. 40. 23. nan 34. 45. 18. 22. 32. 37. 50. 21. 46. 36. 26. 41.
 24. 25. 43. 19. 38. 51. 31. 44. 33. 17.  0. 54.] : 30

Salary : columns [  nan 2300. 3000. 4000. 5000. 6000. 7000. 8000. 9000. 1089. 1234. 3030.
 3045. 3184. 4824. 5835. 7084. 8943. 8345. 9284. 9876. 2034. 7654. 2934.
 4034. 5034. 8202. 9024. 4345. 6544. 6543. 3234. 4324. 5435. 5555. 8787.
 3454. 5654. 5009. 5098. 3033.] : 41

Place : columns ['Chennai' 'Mumbai' 'Calcutta' 'Delhi' 'Podicherry' 'Cochin' nan 'Noida'
 'Hyderabad' 'Bhopal' 'Nagpur' 'Pune'] : 12

Country : columns ['India'] : 1

Gender : columns [0 1] : 2
```

In [7]:
```
# Statistical analysis and renaming of the columns
```

In [8]:
```
df.describe()
```

Out[8]:

	Age	Salary	Gender
count	130.000000	124.000000	148.000000
mean	30.484615	5312.467742	0.222973
std	11.096640	2573.764683	0.417654
min	0.000000	1089.000000	0.000000
25%	22.000000	3030.000000	0.000000
50%	32.500000	5000.000000	0.000000
75%	37.750000	8000.000000	0.000000
max	54.000000	9876.000000	1.000000

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Company     8
Age        18
Salary     24
Place      14
Country     0
Gender      0
dtype: int64
```

In [10]:
```
df.value_counts()
```

Out[10]:

Company	Age	Salary	Place	Country	Gender	
Infosys	22.0	8787.0	Calcutta	India	1	2
TCS	21.0	4824.0	Mumbai	India	0	2
CTS	0.0	1234.0	Calcutta	India	0	1
TCS	0.0	9024.0	Chennai	India	1	1
	23.0	4824.0	Calcutta	India	0	1
						..
Infosys	22.0	8202.0	Cochin	India	0	1
	21.0	3030.0	Calcutta	India	0	1
	0.0	3234.0	Mumbai	India	0	1
		3030.0	Calcutta	India	0	1
Tata Consultancy Services	31.0	8345.0	Mumbai	India	0	1

Name: count, Length: 91, dtype: int64

In [11]:
```
len(df)
```

Out[11]: 148

In [12]:
```
# renamed place to city
df.rename(columns={'Place':'City'}, inplace=True)
df
```

Out[12]:

	Company	Age	Salary	City	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
143	TCS	33.0	9024.0	Calcutta	India	1
144	Infosys	22.0	8787.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

148 rows × 6 columns

DATA CLEANING

In [13]:
```
# missing and inappropriate values
```

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
Company     8
Age        18
Salary     24
City       14
Country     0
Gender      0
dtype: int64
```

In [16]:
```
df['Age'].fillna(df['Age'].median(), inplace=True)
df['Salary'].fillna(df['Salary'].median(), inplace=True)
df['City'].fillna(df['City'].mode()[0], inplace=True)
df['Company'].fillna(df['Company'].mode()[0], inplace=True)
```

In [17]:
```
len(df)
```

Out[17]: 148

In [18]:
```
df.isnull().sum()
```

Out[18]:
```
Company    0
Age        0
Salary     0
City       0
Country    0
Gender     0
dtype: int64
```

In [22]:
```
#df.dropna( inplace=True)
```

In [19]:
```
# df.isnull().sum()
```

In [24]:
```
# Remove all duplicates rows
```

In [20]:
```
df.duplicated().sum()
```

Out[20]: 4

In [21]:
```
df.drop_duplicates(inplace=True)
```

In [22]:
```
df.duplicated().sum()
```

Out[22]: 0

In [23]:
```
# find the outliers
```

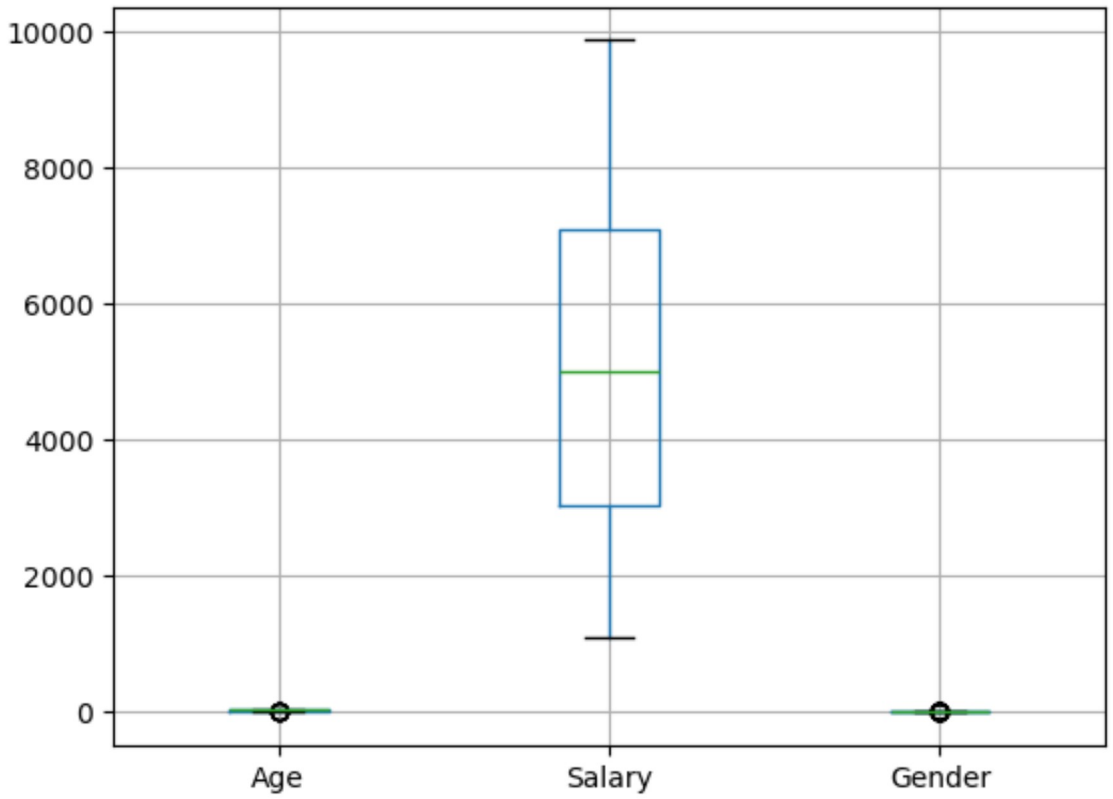
In [24]:
```
nu=df.select_dtypes('number')
```

In [25]:
```
nu.skew()
```

Out[25]:
```
Age      -0.711660
Salary    0.258934
Gender    1.350414
dtype: float64
```

In [26]:
```
nu.boxplot()
```

Out[26]: <Axes: >



In [27]:
```
# Replace the value 0 in age as NaN
```

In [28]:
```
df['Age'].unique()
```

Out[28]:
```
array([20. , 30. , 35. , 40. , 23. , 32.5, 34. , 45. , 18. , 22. , 32. ,
       37. , 50. , 21. , 46. , 36. , 26. , 41. , 24. , 25. , 43. , 19. ,
       38. , 51. , 31. , 44. , 33. , 17. ,  0. , 54. ])
```

In [29]:
```
df['Age'].replace(0,pd.NA , inplace=True)
```

In [30]:
```
df['Age'].unique()
```

Out[30]:
```
array([20.0, 30.0, 35.0, 40.0, 23.0, 32.5, 34.0, 45.0, 18.0, 22.0, 32.0,
       37.0, 50.0, 21.0, 46.0, 36.0, 26.0, 41.0, 24.0, 25.0, 43.0, 19.0,
       38.0, 51.0, 31.0, 44.0, 33.0, 17.0, <NA>, 54.0], dtype=object)
```

DATA ANALYSIS

In [31]:
```
# Filter the data with age >40 and salary< 5000
```

In [32]:
```
any = df[(df['Age']>40) & (df['Salary']<5000)]
any.sample(10)
```

Out[32]:

	Company	Age	Salary	City	Country	Gender
93	Infosys	54.0	3184.0	Mumbai	India	0
21	Infosys	50.0	3184.0	Delhi	India	0
32	Infosys	45.0	4034.0	Calcutta	India	0
138	CTS	44.0	3033.0	Cochin	India	0
104	Infosys	44.0	4034.0	Delhi	India	0
57	Infosys	51.0	3184.0	Hyderabad	India	0
39	Infosys	41.0	3000.0	Mumbai	India	0
122	Infosys	44.0	3234.0	Mumbai	India	0
140	Infosys	44.0	4034.0	Hyderabad	India	0
129	Infosys	50.0	3184.0	Calcutta	India	0

In [33]:
```
# Plot the chart with age and salary
```

In [38]:
```
u=df['Age']
v=df['Salary']
```

In [37]:
```
plt.scatter(u,v)
plt.title('Plot Age & Salary')
plt.xlabel('Age')
plt.ylabel('Salary')
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[37], line 1
----> 1 plt.scatter(u,v)
      2 plt.title('Plot Age & Salary')
      3 plt.xlabel('Age')

File ~\anaconda3\Lib\site-packages\matplotlib\pyplot.py:3687, in scatter(x, y, s, c, marker, cmap, norm, vmin, vmax,
alpha, linewidths, edgecolors, plotnonfinite, data, **kwargs)
   3668 @_copy_docstring_and_deprecators(Axes.scatter)
   3669 def scatter(
   3670     x: float | ArrayLike,
   (...)
   3685     **kwargs,
   3686 ) -> PathCollection:
-> 3687     __ret = gca().scatter(
   3688         x,
   3689         y,
   3690         s=s,
   3691         c=c,
   3692         marker=marker,
   3693         cmap=cmap,
   3694         norm=norm,
   3695         vmin=vmin,
   3696         vmax=vmax,
   3697         alpha=alpha,
   3698         linewidths=linewidths,
   3699         edgecolors=edgecolors,
   3700         plotnonfinite=plotnonfinite,
   3701         **({"data": data} if data is not None else {}),
   3702         **kwargs,
   3703     )
   3704     sci(__ret)
   3705     return __ret

File ~\anaconda3\Lib\site-packages\matplotlib\__init__.py:1465, in _preprocess_data.<locals>.inner(ax, data, *args, *
*kwargs)
   1462 @functools.wraps(func)
   1463 def inner(ax, *args, data=None, **kwargs):
   1464     if data is None:
-> 1465         return func(ax, *map(sanitize_sequence, args), **kwargs)
   1467     bound = new_sig.bind(ax, *args, **kwargs)
```

```
   1468     auto_label = (bound.arguments.get(label_namer)
   1469                   or bound.kwargs.get(label_namer))

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:4741, in Axes.scatter(self, x, y, s, c, marker, cmap, nor
m, vmin, vmax, alpha, linewidths, edgecolors, plotnonfinite, **kwargs)
   4735         linewidths = [
   4736             lw if lw is not None else mpl.rcParams['lines.linewidth']
   4737             for lw in linewidths]
   4739 offsets = np.ma.column_stack([x, y])
-> 4741 collection = mcoll.PathCollection(
   4742     (path,), scales,
   4743     facecolors=colors,
   4744     edgecolors=edgecolors,
   4745     linewidths=linewidths,
   4746     offsets=offsets,
   4747     offset_transform=kwargs.pop('transform', self.transData),
   4748     alpha=alpha,
   4749 )
   4750 collection.set_transform(mtransforms.IdentityTransform())
   4751 if colors is None:

File ~\anaconda3\Lib\site-packages\matplotlib\collections.py:1027, in PathCollection.__init__(self, paths, sizes, **k
wargs)
   1013 def __init__(self, paths, sizes=None, **kwargs):
   1014     """
   1015     Parameters
   1016     ----------
   (...)
   1024         Forwarded to `.Collection`.
   1025     """
-> 1027     super().__init__(**kwargs)
   1028     self.set_paths(paths)
   1029     self.set_sizes(sizes)

File ~\anaconda3\Lib\site-packages\matplotlib\collections.py:194, in Collection.__init__(self, edgecolors, facecolors
, linewidths, linestyles, capstyle, joinstyle, antialiaseds, offsets, offset_transform, norm, cmap, pickradius, hatch
, urls, zorder, **kwargs)
    191     self._joinstyle = None
    193 if offsets is not None:
--> 194     offsets = np.asanyarray(offsets, float)
    195     # Broadcast (2,) -> (1, 2) but nothing else.
    196     if offsets.shape == (2,):
```

```
TypeError: float() argument must be a string or a real number, not 'NAType'
```

In [ ]:
```
# Count the number of people from each place and represent it visullay
```

In [ ]:
```
plc=df['City'].value_counts()
plc
# value-count is used for char
```

In [36]:
```
plt.pie(plc.values, labels=plc.index,  autopct='%1.1f%%')
plt.figure(figsize=(19,19))
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[36], line 1
----> 1 plt.pie(plc.values, labels=plc.index,  autopct='%1.1f%%')
      2 plt.figure(figsize=(19,19))
      3 plt.show()

NameError: name 'plc' is not defined
```

DATA ENCODING

In [ ]:
```
# convert categorical variable into numerical
```

In [ ]:
```
df
```

In [ ]:
```
cvt = pd.get_dummies(df,columns=['Company', 'Place', 'Country'])
cvt
```

In [59]:
```
from sklearn.preprocessing import LabelEncoder

labelencode = LabelEncoder()
df['City'] = labelencode.fit_transform(df['City'])
df.sample(10)
```

Out[59]:

	Company	Age	Salary	City	Country	Gender
65	CTS	31.000000	2034.000000	5	0	0
102	CTS	44.000000	5312.467742	6	0	0
11	CTS	34.000000	1089.000000	1	0	0
38	TCS	34.000000	2300.000000	1	0	0
32	Infosys	45.000000	4034.000000	1	0	0
92	TCS	32.000000	3045.000000	4	0	0
93	Infosys	54.000000	3184.000000	6	0	0
104	Infosys	44.000000	4034.000000	4	0	0
23	Infosys	30.484615	5835.000000	3	0	0
94	TCS	22.000000	4824.000000	1	0	0

FEATURE SCALING

In [59]:
```
# Standardscaler & MinMaxScaler
```

In [71]:
```
from sklearn.preprocessing import StandardScaler , MinMaxScaler
```

In [75]:
```
z = ['Age', 'Salary']
```

In [77]:
```
sc = StandardScaler()
a = sc.fit_transform(df[z])
a
```

Out[77]:
```
array([[-1.42789959,  0.10870042],
       [-0.24940097,  0.10870042],
       [ 0.33984834, -1.1783297 ],
       [ 0.92909765, -0.87926556],
       [-1.07435   , -0.45203107],
       [-0.19228911, -0.02479658],
       [-0.19228911,  0.40243791],
       [-1.07435   ,  0.8296724 ],
       [ 0.22199848,  1.25690689],
       [ 1.51834697,  1.68414138],
       [-1.07435   ,  0.10870042],
       [ 0.22199848, -1.69571067],
       [ 1.51834697,  0.10870042],
       [-1.66359932, -1.63376166],
       [ 0.92909765, -0.87926556],
       [-1.07435   , -0.87926556],
       [-1.07435   , -0.86644852],
       [ 0.22199848, -0.02479658],
       [-1.19219987,  0.10870042],
       [-0.01370124,  0.10870042],
       [ 0.57554807, -0.86004   ],
       [ 2.10759628, -0.80065441],
       [-1.31004973, -0.09998985],
       [-0.19228911,  0.33194422],
       [-0.19228911,  0.8655601 ],
       [-1.07435   ,  1.65978902],
       [ 0.22199848,  1.40430279],
       [ 1.51834697,  1.80547598],
       [-1.07435   ,  2.0583988 ],
       [ 0.33984834, -1.29197407],
       [ 1.63619683,  1.10908376],
       [-1.42789959, -0.90746303],
       [ 1.51834697, -0.43750509],
       [-0.01370124,  0.10870042],
       [ 0.22199848, -1.1783297 ],
       [ 1.04694752, -0.87926556],
       [-0.95650014, -0.30463517],
       [ 0.33984834,  1.25690689],
       [ 1.63619683,  1.68414138],
       [-0.95650014,  0.10870042],
       [-0.01370124, -1.69571067],
       [ 1.28264724,  0.10870042],
```

```
[-1.54574945, -1.63376166],
[ 1.04694752, -0.87926556],
[-1.31004973, -0.86644852],
[ 0.33984834, -0.02479658],
[-1.31004973,  0.63485348],
[-0.01370124,  1.10908376],
[ 0.69339793, -0.86004   ],
[ 2.22544614, -0.80065441],
[-1.07435   , -0.09998985],
[-0.19228911,  0.33194422],
[-0.19228911,  0.8655601 ],
[-0.13155111, -1.29197407],
[ 1.04694752,  0.10870042],
[-1.31004973, -0.90746303],
[ 1.28264724, -0.43750509],
[-0.01370124, -0.0102706 ],
[-1.31004973,  1.34320826],
[ 0.22199848,  1.69439501],
[-0.95650014,  0.10870042],
[ 0.22199848,  0.10870042],
[ 0.57554807, -1.1783297 ],
[ 1.4004971 , -0.87926556],
[-0.01370124, -0.45203107],
[-0.19228911, -0.02479658],
[-0.19228911,  0.40243791],
[-0.72080042,  0.8296724 ],
[-0.01370124,  1.25690689],
[ 1.28264724,  1.68414138],
[-1.19219987,  0.10870042],
[ 0.10414862, -1.69571067],
[-1.78144918, -1.63376166],
[ 1.04694752, -0.87926556],
[-1.31004973, -0.87926556],
[-0.01370124, -0.86644852],
[ 0.45769821, -0.02479658],
[-1.31004973,  0.10870042],
[ 0.22199848,  0.10870042],
[-0.01370124, -0.86004   ],
[ 2.57899573, -0.80065441],
[-1.19219987, -0.09998985],
[-0.19228911,  0.33194422],
[-0.19228911,  0.8655601 ],
```

```
       [-1.19219987,  1.65978902],
       [ 0.10414862,  1.40430279],
       [ 1.4004971 ,  1.80547598],
       [-1.19219987,  0.10870042],
       [-0.01370124, -1.29197407],
       [ 1.4004971 ,  0.10870042],
       [-1.19219987, -0.90746303],
       [ 1.4004971 , -0.43750509],
       [ 0.10414862, -0.0102706 ],
       [-1.19219987,  1.34320826],
       [-0.01370124,  1.69439501],
       [ 0.10414862,  0.10870042],
       [ 0.10414862, -1.1783297 ],
       [-0.01370124, -0.77929268],
       [-1.19219987, -0.31360709],
       [-0.19228911,  0.16105043],
       [-0.19228911,  0.21231857],
       [-1.19219987,  1.59314044],
       [ 0.10414862, -1.69571067],
       [ 1.4004971 ,  0.10870042],
       [-0.01370124, -1.63376166],
       [ 1.4004971 , -0.77929268],
       [-1.19219987,  1.59314044],
       [-0.24940097,  0.0170724 ],
       [ 0.10414862, -0.86004   ],
       [ 2.10759628, -0.80065441],
       [-0.19228911,  0.8655601 ],
       [-0.13155111,  1.40430279],
       [ 0.92909765,  1.80547598],
       [-0.13155111, -1.29197407],
       [ 1.4004971 , -0.86516682],
       [-1.19219987, -0.90746303],
       [ 1.4004971 , -0.43750509],
       [ 0.10414862, -0.0102706 ],
       [-1.19219987,  1.34320826],
       [ 0.10414862,  1.69439501],
       [ 1.4004971 , -0.43750509],
       [ 0.10414862, -0.0102706 ],
       [-1.19219987,  1.34320826]])
```

In [74]:
```
ms = MinMaxScaler()
b= ms.fit_transform(df[z])
```

```
b
```

Out[74]:
```
array([[0.08108108, 0.48064957],
       [0.35135135, 0.48064957],
       [0.48648649, 0.13781723],
       [0.62162162, 0.21748037],
       [0.16216216, 0.33128485],
       [0.36444906, 0.44508934],
       [0.36444906, 0.55889382],
       [0.16216216, 0.6726983 ],
       [0.45945946, 0.78650279],
       [0.75675676, 0.90030727],
       [0.16216216, 0.48064957],
       [0.45945946, 0.        ],
       [0.75675676, 0.48064957],
       [0.02702703, 0.01650165],
       [0.62162162, 0.21748037],
       [0.16216216, 0.21748037],
       [0.16216216, 0.2208945 ],
       [0.45945946, 0.44508934],
       [0.13513514, 0.48064957],
       [0.40540541, 0.48064957],
       [0.54054054, 0.22260157],
       [0.89189189, 0.23842039],
       [0.10810811, 0.42505975],
       [0.36444906, 0.54011608],
       [0.36444906, 0.68225788],
       [0.16216216, 0.89382042],
       [0.45945946, 0.82576534],
       [0.75675676, 0.93262775],
       [0.16216216, 1.        ],
       [0.48648649, 0.10754524],
       [0.78378378, 0.74712644],
       [0.08108108, 0.20996927],
       [0.75675676, 0.33515421],
       [0.40540541, 0.48064957],
       [0.45945946, 0.13781723],
       [0.64864865, 0.21748037],
       [0.18918919, 0.3705474 ],
       [0.48648649, 0.78650279],
       [0.78378378, 0.90030727],
       [0.18918919, 0.48064957],
       [0.40540541, 0.        ],
       [0.7027027 , 0.48064957],
```

```
[0.05405405, 0.01650165],
[0.64864865, 0.21748037],
[0.10810811, 0.2208945 ],
[0.48648649, 0.44508934],
[0.10810811, 0.62080346],
[0.40540541, 0.74712644],
[0.56756757, 0.22260157],
[0.91891892, 0.23842039],
[0.16216216, 0.42505975],
[0.36444906, 0.54011608],
[0.36444906, 0.68225788],
[0.37837838, 0.10754524],
[0.64864865, 0.48064957],
[0.10810811, 0.20996927],
[0.7027027 , 0.33515421],
[0.40540541, 0.44895869],
[0.10810811, 0.80949129],
[0.45945946, 0.90303858],
[0.18918919, 0.48064957],
[0.45945946, 0.48064957],
[0.54054054, 0.13781723],
[0.72972973, 0.21748037],
[0.40540541, 0.33128485],
[0.36444906, 0.44508934],
[0.36444906, 0.55889382],
[0.24324324, 0.6726983 ],
[0.40540541, 0.78650279],
[0.7027027 , 0.90030727],
[0.13513514, 0.48064957],
[0.43243243, 0.        ],
[0.        , 0.01650165],
[0.64864865, 0.21748037],
[0.10810811, 0.21748037],
[0.40540541, 0.2208945 ],
[0.51351351, 0.44508934],
[0.10810811, 0.48064957],
[0.45945946, 0.48064957],
[0.40540541, 0.22260157],
[1.        , 0.23842039],
[0.13513514, 0.42505975],
[0.36444906, 0.54011608],
[0.36444906, 0.68225788],
```

```
       [0.13513514, 0.89382042],
       [0.43243243, 0.82576534],
       [0.72972973, 0.93262775],
       [0.13513514, 0.48064957],
       [0.40540541, 0.10754524],
       [0.72972973, 0.48064957],
       [0.13513514, 0.20996927],
       [0.72972973, 0.33515421],
       [0.43243243, 0.44895869],
       [0.13513514, 0.80949129],
       [0.40540541, 0.90303858],
       [0.43243243, 0.48064957],
       [0.43243243, 0.13781723],
       [0.40540541, 0.24411062],
       [0.13513514, 0.36815751],
       [0.36444906, 0.49459429],
       [0.36444906, 0.50825083],
       [0.13513514, 0.87606692],
       [0.43243243, 0.        ],
       [0.72972973, 0.48064957],
       [0.40540541, 0.01650165],
       [0.72972973, 0.24411062],
       [0.13513514, 0.87606692],
       [0.35135135, 0.45624218],
       [0.43243243, 0.22260157],
       [0.89189189, 0.23842039],
       [0.36444906, 0.68225788],
       [0.37837838, 0.82576534],
       [0.62162162, 0.93262775],
       [0.37837838, 0.10754524],
       [0.72972973, 0.22123592],
       [0.13513514, 0.20996927],
       [0.72972973, 0.33515421],
       [0.43243243, 0.44895869],
       [0.13513514, 0.80949129],
       [0.43243243, 0.90303858],
       [0.72972973, 0.33515421],
       [0.43243243, 0.44895869],
       [0.13513514, 0.80949129]])
```

In [ ]: