

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: df = pd.read_csv('titanic_dataset.csv')
```

```
In [3]: df
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
	0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
	1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
	2	3	1	3	Heikinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
	3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
	4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
	...	...	...	...	...	...	...	...	...	...	...	...	...
	886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
	887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
	888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
	889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
	890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

```
In [7]: df.isnull().sum()
```

Out[7]:

PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2
dtype:	int64

```
In [11]: df1=df.drop(['Embarked','Cabin','Ticket','Name'],axis=1)
df1
```

Out[11]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	
	0	1	0	3	male	22.0	1	0	7.2500
	1	2	1	1	female	38.0	1	0	71.2833
	2	3	1	3	female	26.0	0	0	7.9250
	3	4	1	1	female	35.0	1	0	53.1000
	4	5	0	3	male	35.0	0	0	8.0500
	...	...	...	...	...	...	...	...	...
	886	887	0	2	male	27.0	0	0	13.0000
	887	888	1	1	female	19.0	0	0	30.0000
	888	889	0	3	female	NaN	1	2	23.4500
	889	890	1	1	male	26.0	0	0	30.0000
	890	891	0	3	male	32.0	0	0	7.7500

891 rows × 8 columns

In [13]:

```
df1['Age'].fillna(0,inplace=True)
df1
```

Out[13]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	
	0	1	0	3	male	22.0	1	0	7.2500
	1	2	1	1	female	38.0	1	0	71.2833
	2	3	1	3	female	26.0	0	0	7.9250
	3	4	1	1	female	35.0	1	0	53.1000
	4	5	0	3	male	35.0	0	0	8.0500
	...	...	...	...	...	...	...	...	...
	886	887	0	2	male	27.0	0	0	13.0000
	887	888	1	1	female	19.0	0	0	30.0000
	888	889	0	3	female	0.0	1	2	23.4500
	889	890	1	1	male	26.0	0	0	30.0000
	890	891	0	3	male	32.0	0	0	7.7500

891 rows × 8 columns

In [15]:

```
v = df1.groupby('Pclass')['Pclass'].count() #group by function use to group count
v
```

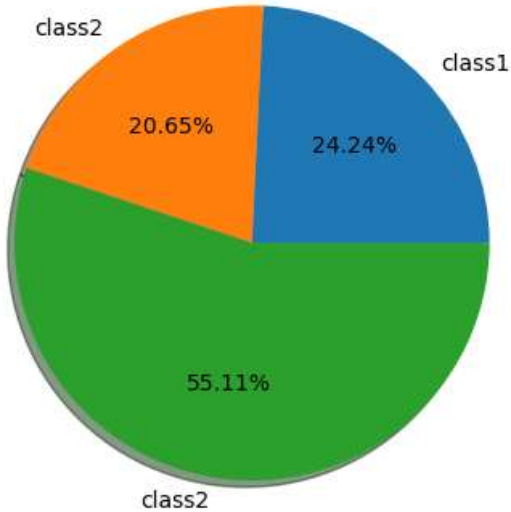
```
Out[15]: Pclass
1      216
2      184
3      491
Name: Pclass, dtype: int64
```

```
In [17]: from matplotlib import pyplot as plt
```

```
In [19]: v.values
```

```
Out[19]: array([216, 184, 491], dtype=int64)
```

```
In [29]: mylabels = ['class1','class2','class2']
plt.pie(v, labels = mylabels,shadow=True,autopct='%1.2f%%') #atopct for displaying float percentile on chart
plt.show()
```



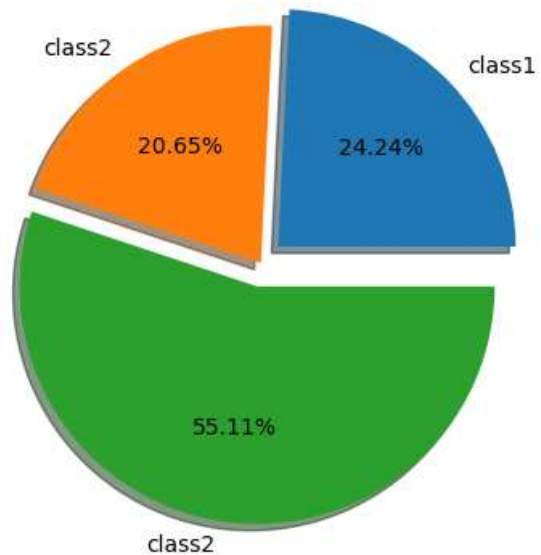
```
In [28]: df1.loc[1:5]
```

```
Out[28]:
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
1	2	1	1	female	38.0	1	0	71.2833
2	3	1	3	female	26.0	0	0	7.9250
3	4	1	1	female	35.0	1	0	53.1000
4	5	0	3	male	35.0	0	0	8.0500
5	6	0	3	male	0.0	0	0	8.4583

```
In [32]: mylabels = ['class1','class2','class2']
explode = (0.1,0.0,0.1) #for seperation
```

```
plt.pie(v, labels = mylabels,shadow=True,explode = explode,autopct='%1.2f%%') #atopct for displaying float percentile on chart
plt.show()
```



```
In [34]: df1.isnull().sum()
```

```
Out[34]: PassengerId    0
Survived      0
Pclass        0
Sex           0
Age           0
SibSp         0
Parch         0
Fare          0
dtype: int64
```

```
In [39]: a = df1.groupby(['Sex','Pclass'])['Age'].mean()
a
```

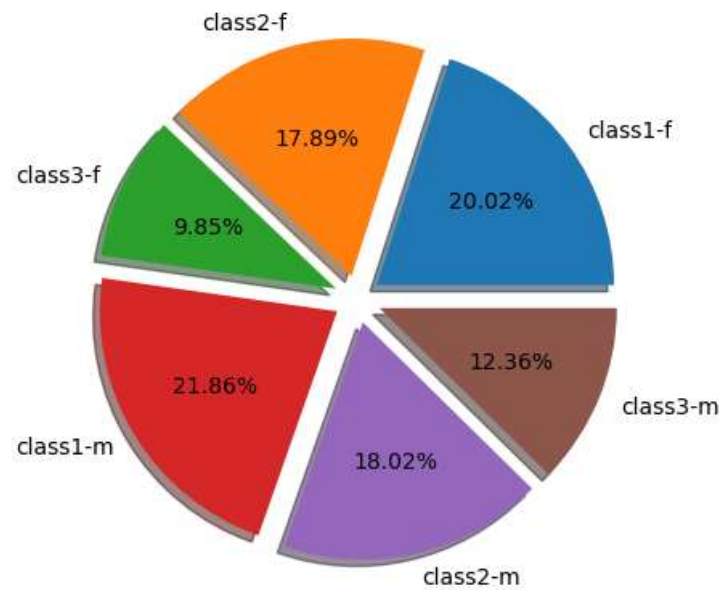
```
Out[39]: Sex    Pclass
female  1      31.297872
         2      27.967105
         3      15.406250
male    1      34.175574
         2      28.178981
         3      19.326859
Name: Age, dtype: float64
```

```
In [41]: a.values
```

```
Out[41]: array([31.29787234, 27.96710526, 15.40625    , 34.17557377, 28.17898148,
        19.32685879])
```

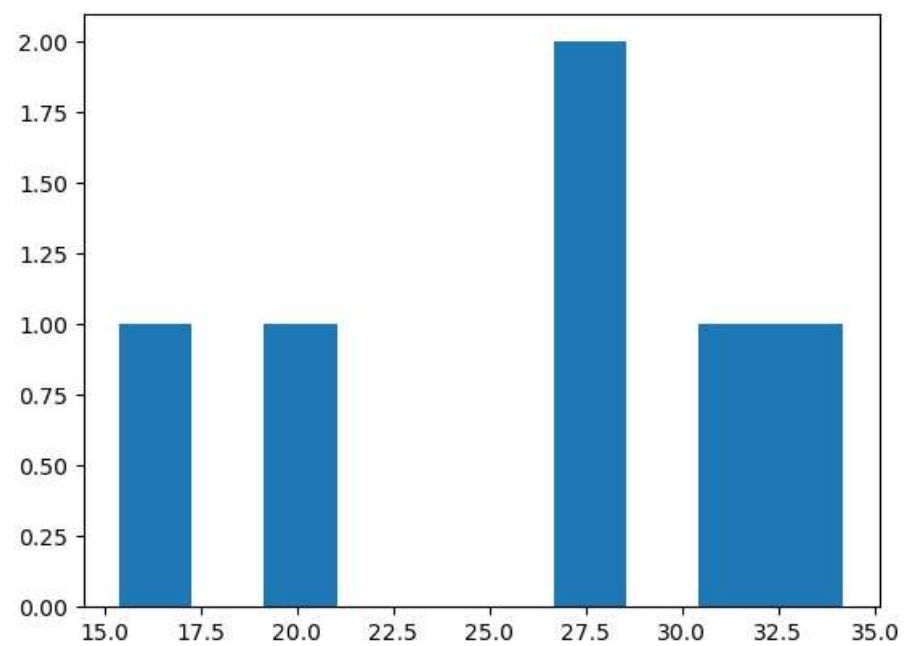
```
In [51]: mylabel1 = ['class1-f','class2-f','class3-f','class1-m','class2-m','class3-m']
explode = (0.1,0.1,0.1,0.1,0.1,0.1) #for seperation of pizza slices
```

```
plt.pie(a, labels = mylabel1,shadow=True,explode = explode,autopct='%1.2f%%') #atopct for displaying float percentile on chart
plt.show()
```



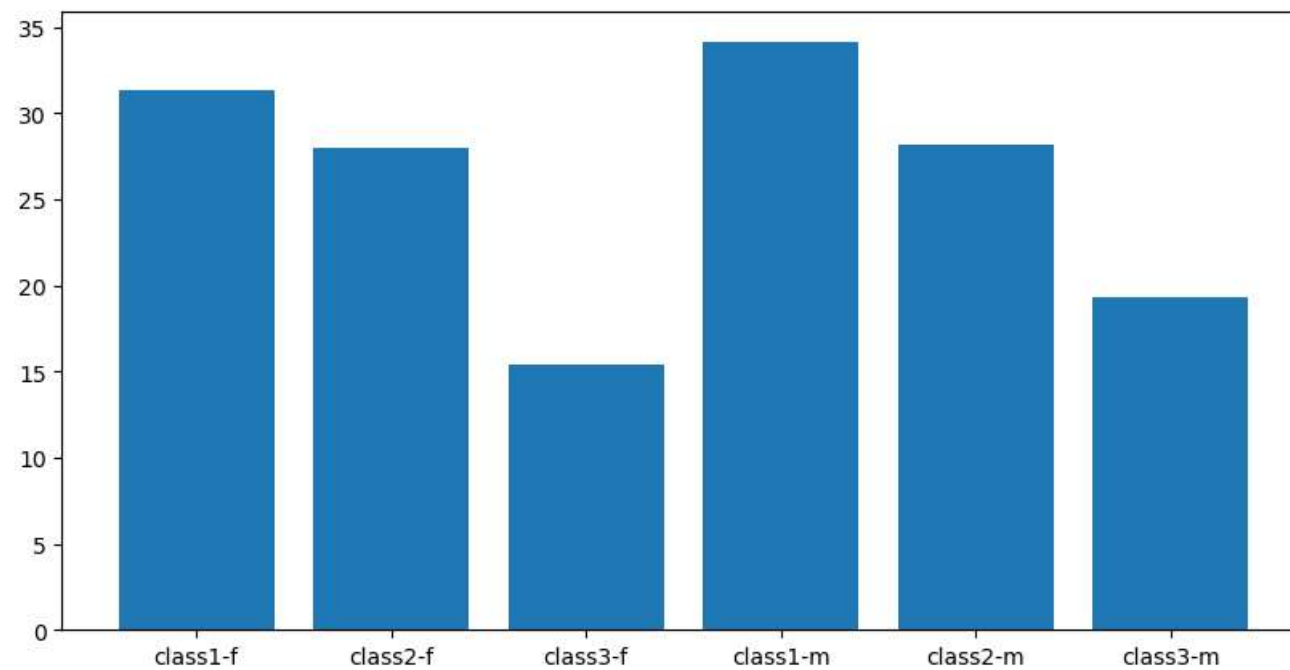
```
In [54]: plt.hist(a)
plt.show
```

```
Out[54]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
In [58]: fig = plt.figure(figsize=(10,5))  
plt.bar(mylabel1,a)
```

```
Out[58]: <BarContainer object of 6 artists>
```



```
In [57]: df1['Fare'].notnull().sum()
```

```
Out[57]: 891
```

```
In [59]: fare=df1[df['Fare'].notnull()]['Fare'].values  
fare
```

```
Out[59]: array([ 7.25 , 71.2833, 7.925 , 53.1 , 8.05 , 8.4583,
51.8625, 21.075 , 11.1333, 30.0708, 16.7 , 26.55 ,
8.05 , 31.275 , 7.8542, 16. , 29.125 , 13. ,
18. , 7.225 , 26. , 13. , 8.0292, 35.5 ,
21.075 , 31.3875, 7.225 , 263. , 7.8792, 7.8958,
27.7208, 146.5208, 7.75 , 10.5 , 82.1708, 52. ,
7.2292, 8.05 , 18. , 11.2417, 9.475 , 21. ,
7.8958, 41.5792, 7.8792, 8.05 , 15.5 , 7.75 ,
21.6792, 17.8 , 39.6875, 7.8 , 76.7292, 26. ,
61.9792, 35.5 , 10.5 , 7.2292, 27.75 , 46.9 ,
7.2292, 80. , 83.475 , 27.9 , 27.7208, 15.2458,
10.5 , 8.1583, 7.925 , 8.6625, 10.5 , 46.9 ,
73.5 , 14.4542, 56.4958, 7.65 , 7.8958, 8.05 ,
29. , 12.475 , 9. , 9.5 , 7.7875, 47.1 ,
10.5 , 15.85 , 34.375 , 8.05 , 263. , 8.05 ,
8.05 , 7.8542, 61.175 , 20.575 , 7.25 , 8.05 ,
34.6542, 63.3583, 23. , 26. , 7.8958, 7.8958,
77.2875, 8.6542, 7.925 , 7.8958, 7.65 , 7.775 ,
7.8958, 24.15 , 52. , 14.4542, 8.05 , 9.825 ,
14.4583, 7.925 , 7.75 , 21. , 247.5208, 31.275 ,
73.5 , 8.05 , 30.0708, 13. , 77.2875, 11.2417,
7.75 , 7.1417, 22.3583, 6.975 , 7.8958, 7.05 ,
14.5 , 26. , 13. , 15.0458, 26.2833, 53.1 ,
9.2167, 79.2 , 15.2458, 7.75 , 15.85 , 6.75 ,
11.5 , 36.75 , 7.7958, 34.375 , 26. , 13. ,
12.525 , 66.6 , 8.05 , 14.5 , 7.3125, 61.3792,
7.7333, 8.05 , 8.6625, 69.55 , 16.1 , 15.75 ,
7.775 , 8.6625, 39.6875, 20.525 , 55. , 27.9 ,
25.925 , 56.4958, 33.5 , 29.125 , 11.1333, 7.925 ,
30.6958, 7.8542, 25.4667, 28.7125, 13. , 0. ,
69.55 , 15.05 , 31.3875, 39. , 22.025 , 50. ,
15.5 , 26.55 , 15.5 , 7.8958, 13. , 13. ,
7.8542, 26. , 27.7208, 146.5208, 7.75 , 8.4042,
7.75 , 13. , 9.5 , 69.55 , 6.4958, 7.225 ,
8.05 , 10.4625, 15.85 , 18.7875, 7.75 , 31. ,
7.05 , 21. , 7.25 , 13. , 7.75 , 113.275 ,
7.925 , 27. , 76.2917, 10.5 , 8.05 , 13. ,
8.05 , 7.8958, 90. , 9.35 , 10.5 , 7.25 ,
13. , 25.4667, 83.475 , 7.775 , 13.5 , 31.3875,
10.5 , 7.55 , 26. , 26.25 , 10.5 , 12.275 ,
14.4542, 15.5 , 10.5 , 7.125 , 7.225 , 90. ,
7.775 , 14.5 , 52.5542, 26. , 7.25 , 10.4625,
26.55 , 16.1 , 20.2125, 15.2458, 79.2 , 86.5 ,
512.3292, 26. , 7.75 , 31.3875, 79.65 , 0. ,
7.75 , 10.5 , 39.6875, 7.775 , 153.4625, 135.6333,
31. , 0. , 19.5 , 29.7 , 7.75 , 77.9583,
7.75 , 0. , 29.125 , 20.25 , 7.75 , 7.8542,
9.5 , 8.05 , 26. , 8.6625, 9.5 , 7.8958,
13. , 7.75 , 78.85 , 91.0792, 12.875 , 8.85 ,
7.8958, 27.7208, 7.2292, 151.55 , 30.5 , 247.5208,
7.75 , 23.25 , 0. , 12.35 , 8.05 , 151.55 ,
110.8833, 108.9 , 24. , 56.9292, 83.1583, 262.375 ,
26. , 7.8958, 26.25 , 7.8542, 26. , 14. ,
164.8667, 134.5 , 7.25 , 7.8958, 12.35 , 29. ,
69.55 , 135.6333, 6.2375, 13. , 20.525 , 57.9792,
23.25 , 28.5 , 153.4625, 18. , 133.65 , 7.8958,
66.6 , 134.5 , 8.05 , 35.5 , 26. , 263. ,
13. , 13. , 13. , 13. , 13. , 16.1 ,
```



15.9 , 8.6625, 9.225 , 35. , 7.2292, 17.8 ,  
7.225 , 9.5 , 55. , 13. , 7.8792, 7.8792,  
27.9 , 27.7208, 14.4542, 7.05 , 15.5 , 7.25 ,  
75.25 , 7.2292, 7.75 , 69.3 , 55.4417, 6.4958,  
8.05 , 135.6333, 21.075 , 82.1708, 7.25 , 211.5 ,  
4.0125, 7.775 , 227.525 , 15.7417, 7.925 , 52. ,  
7.8958, 73.5 , 46.9 , 13. , 7.7292, 12. ,  
120. , 7.7958, 7.925 , 113.275 , 16.7 , 7.7958,  
7.8542, 26. , 10.5 , 12.65 , 7.925 , 8.05 ,  
9.825 , 15.85 , 8.6625, 21. , 7.75 , 18.75 ,  
7.775 , 25.4667, 7.8958, 6.8583, 90. , 0. ,  
7.925 , 8.05 , 32.5 , 13. , 13. , 24.15 ,  
7.8958, 7.7333, 7.875 , 14.4 , 20.2125, 7.25 ,  
26. , 26. , 7.75 , 8.05 , 26.55 , 16.1 ,  
26. , 7.125 , 55.9 , 120. , 34.375 , 18.75 ,  
263. , 10.5 , 26.25 , 9.5 , 7.775 , 13. ,  
8.1125, 81.8583, 19.5 , 26.55 , 19.2583, 30.5 ,  
27.75 , 19.9667, 27.75 , 89.1042, 8.05 , 7.8958,  
26.55 , 51.8625, 10.5 , 7.75 , 26.55 , 8.05 ,  
38.5 , 13. , 8.05 , 7.05 , 0. , 26.55 ,  
7.725 , 19.2583, 7.25 , 8.6625, 27.75 , 13.7917,  
9.8375, 52. , 21. , 7.0458, 7.5208, 12.2875,  
46.9 , 0. , 8.05 , 9.5875, 91.0792, 25.4667,  
90. , 29.7 , 8.05 , 15.9 , 19.9667, 7.25 ,  
30.5 , 49.5042, 8.05 , 14.4583, 78.2667, 15.1 ,  
151.55 , 7.7958, 8.6625, 7.75 , 7.6292, 9.5875,  
86.5 , 108.9 , 26. , 26.55 , 22.525 , 56.4958,  
7.75 , 8.05 , 26.2875, 59.4 , 7.4958, 34.0208,  
10.5 , 24.15 , 26. , 7.8958, 93.5 , 7.8958,  
7.225 , 57.9792, 7.2292, 7.75 , 10.5 , 221.7792,  
7.925 , 11.5 , 26. , 7.2292, 7.2292, 22.3583,  
8.6625, 26.25 , 26.55 , 106.425 , 14.5 , 49.5 ,  
71. , 31.275 , 31.275 , 26. , 106.425 , 26. ,  
26. , 13.8625, 20.525 , 36.75 , 110.8833, 26. ,  
7.8292, 7.225 , 7.775 , 26.55 , 39.6 , 227.525 ,  
79.65 , 17.4 , 7.75 , 7.8958, 13.5 , 8.05 ,  
8.05 , 24.15 , 7.8958, 21.075 , 7.2292, 7.8542,  
10.5 , 51.4792, 26.3875, 7.75 , 8.05 , 14.5 ,  
13. , 55.9 , 14.4583, 7.925 , 30. , 110.8833,  
26. , 40.125 , 8.7125, 79.65 , 15. , 79.2 ,  
8.05 , 8.05 , 7.125 , 78.2667, 7.25 , 7.75 ,  
26. , 24.15 , 33. , 0. , 7.225 , 56.9292,  
27. , 7.8958, 42.4 , 8.05 , 26.55 , 15.55 ,  
7.8958, 30.5 , 41.5792, 153.4625, 31.275 , 7.05 ,  
15.5 , 7.75 , 8.05 , 65. , 14.4 , 16.1 ,  
39. , 10.5 , 14.4542, 52.5542, 15.7417, 7.8542,  
16.1 , 32.3208, 12.35 , 77.9583, 7.8958, 7.7333,  
30. , 7.0542, 30.5 , 0. , 27.9 , 13. ,  
7.925 , 26.25 , 39.6875, 16.1 , 7.8542, 69.3 ,  
27.9 , 56.4958, 19.2583, 76.7292, 7.8958, 35.5 ,  
7.55 , 7.55 , 7.8958, 23. , 8.4333, 7.8292,  
6.75 , 73.5 , 7.8958, 15.5 , 13. , 113.275 ,  
133.65 , 7.225 , 25.5875, 7.4958, 7.925 , 73.5 ,  
13. , 7.775 , 8.05 , 52. , 39. , 52. ,  
10.5 , 13. , 0. , 7.775 , 8.05 , 9.8417,  
46.9 , 512.3292, 8.1375, 76.7292, 9.225 , 46.9 ,  
39. , 41.5792, 39.6875, 10.1708, 7.7958, 211.3375,  
57. , 13.4167, 56.4958, 7.225 , 26.55 , 13.5 ,

```

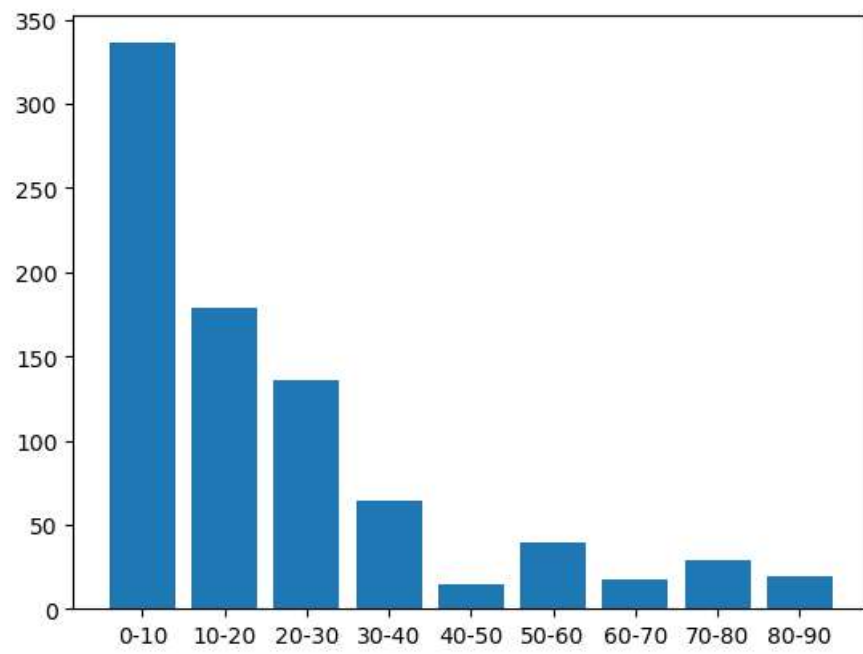
8.05 , 7.7333, 110.8833, 7.65 , 227.525 , 26.2875,
14.4542, 7.7417, 7.8542, 26. , 13.5 , 26.2875,
151.55 , 15.2458, 49.5042, 26.55 , 52. , 9.4833,
13. , 7.65 , 227.525 , 10.5 , 15.5 , 7.775 ,
33. , 7.0542, 13. , 13. , 53.1 , 8.6625,
21. , 7.7375, 26. , 7.925 , 211.3375, 18.7875,
0. , 13. , 13. , 16.1 , 34.375 , 512.3292,
7.8958, 7.8958, 30. , 78.85 , 262.375 , 16.1 ,
7.925 , 71. , 20.25 , 13. , 53.1 , 7.75 ,
23. , 12.475 , 9.5 , 7.8958, 65. , 14.5 ,
7.7958, 11.5 , 8.05 , 86.5 , 14.5 , 7.125 ,
7.2292, 120. , 7.775 , 77.9583, 39.6 , 7.75 ,
24.15 , 8.3625, 9.5 , 7.8542, 10.5 , 7.225 ,
23. , 7.75 , 7.75 , 12.475 , 7.7375, 211.3375,
7.2292, 57. , 30. , 23.45 , 7.05 , 7.25 ,
7.4958, 29.125 , 20.575 , 79.2 , 7.75 , 26. ,
69.55 , 30.6958, 7.8958, 13. , 25.9292, 8.6833,
7.2292, 24.15 , 13. , 26.25 , 120. , 8.5167,
6.975 , 7.775 , 0. , 7.775 , 13. , 53.1 ,
7.8875, 24.15 , 10.5 , 31.275 , 8.05 , 0. ,
7.925 , 37.0042, 6.45 , 27.9 , 93.5 , 8.6625,
0. , 12.475 , 39.6875, 6.95 , 56.4958, 37.0042,
7.75 , 80. , 14.4542, 18.75 , 7.2292, 7.8542,
8.3 , 83.1583, 8.6625, 8.05 , 56.4958, 29.7 ,
7.925 , 10.5 , 31. , 6.4375, 8.6625, 7.55 ,
69.55 , 7.8958, 33. , 89.1042, 31.275 , 7.775 ,
15.2458, 39.4 , 26. , 9.35 , 164.8667, 26.55 ,
19.2583, 7.2292, 14.1083, 11.5 , 25.9292, 69.55 ,
13. , 13. , 13.8583, 50.4958, 9.5 , 11.1333,
7.8958, 52.5542, 5. , 9. , 24. , 7.225 ,
9.8458, 7.8958, 7.8958, 83.1583, 26. , 7.8958,
10.5167, 10.5 , 7.05 , 29.125 , 13. , 30. ,
23.45 , 30. , 7.75 ])
```

```

In [61]: fare_hist = np.histogram(fare,bins = [0,10,20,30,40,50,60,70,80,90])
fare_hist_label = ['0-10','10-20','20-30','30-40','40-50','50-60','60-70','70-80','80-90']
plt.bar(fare_hist_label,fare_hist[0])
```

```

Out[61]: <BarContainer object of 9 artists>
```



In [ ]: