

```
In [13]: import pandas as pd  
import numpy as np  
import random as rnd  
import seaborn as sns  
import matplotlib.pyplot as plt
```

```
In [14]: df=pd.read_csv(r"C:\Users\Shobhit Khaiwal\Downloads\tested.csv")
```

```
In [15]: df
```

Out[15]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298
...
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668

418 rows × 12 columns

In [16]: df.head

```
Out[16]: <bound method NDFrame.head of
   PassengerId  Survived  Pclass
0            892       0      3 \
1            893       1      3
2            894       0      2
3            895       0      3
4            896       1      3
 ..
413          ...     ...
414          1305      0      3
414          1306      1      1
415          1307      0      3
416          1308      0      3
417          1309      0      3

   Name      Sex   Age  SibSp  Parch
0 Kelly, Mr. James    male  34.5      0      0 \
1 Wilkes, Mrs. James (Ellen Needs)  female  47.0      1      0
2 Myles, Mr. Thomas Francis    male  62.0      0      0
3 Wirz, Mr. Albert    male  27.0      0      0
4 Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1      1
..
413 Spector, Mr. Woolf    male   NaN      0      0
414 Oliva y Ocana, Dona. Fermina  female  39.0      0      0
415 Saether, Mr. Simon Sivertsen    male  38.5      0      0
416 Ware, Mr. Frederick    male   NaN      0      0
417 Peter, Master. Michael J    male   NaN      1      1

   Ticket      Fare Cabin Embarked
0      330911  7.8292   NaN      Q
1      363272  7.0000   NaN      S
2      240276  9.6875   NaN      Q
3      315154  8.6625   NaN      S
4      3101298 12.2875   NaN      S
..
413      A.5. 3236  8.0500   NaN      S
414      PC 17758 108.9000  C105      C
415  SOTON/O.Q. 3101262  7.2500   NaN      S
416      359309  8.0500   NaN      S
417      2668  22.3583   NaN      C

[418 rows x 12 columns]>
```

In [17]: df.tail

```
Out[17]: <bound method NDFrame.tail of      PassengerId  Survived  Pclass
0          892        0       3 \
1          893        1       3
2          894        0       2
3          895        0       3
4          896        1       3
...
413        1305       0       3
414        1306       1       1
415        1307       0       3
416        1308       0       3
417        1309       0       3

          Name     Sex   Age  SibSp  Parch
0  Kelly, Mr. James    male  34.5      0      0
\
1  Wilkes, Mrs. James (Ellen Needs)  female  47.0      1      0
2  Myles, Mr. Thomas Francis    male  62.0      0      0
3  Wirz, Mr. Albert    male  27.0      0      0
4  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1      1
...
413  Spector, Mr. Woolf    male   NaN      0      0
414  Oliva y Ocana, Dona. Fermina  female  39.0      0      0
415  Saether, Mr. Simon Sivertsen    male  38.5      0      0
416  Ware, Mr. Frederick    male   NaN      0      0
417  Peter, Master. Michael J    male   NaN      1      1

      Ticket     Fare Cabin Embarked
0      330911  7.8292   NaN      Q
1      363272  7.0000   NaN      S
2      240276  9.6875   NaN      Q
3      315154  8.6625   NaN      S
4      3101298 12.2875   NaN      S
...
413  A.5. 3236  8.0500   NaN      S
414  PC 17758  108.9000  C105      C
415  SOTON/O.Q. 3101262  7.2500   NaN      S
416            359309  8.0500   NaN      S
417            2668  22.3583   NaN      C

[418 rows x 12 columns]>
```

In [18]: df.info

```
Out[18]: <bound method DataFrame.info of
0          892        0      3 \
1          893        1      3
2          894        0      2
3          895        0      3
4          896        1      3
..         ...
413        1305       0      3
414        1306       1      1
415        1307       0      3
416        1308       0      3
417        1309       0      3

Name      Sex   Age  SibSp  Parch
0    Kelly, Mr. James    male  34.5     0     0
\
1  Wilkes, Mrs. James (Ellen Needs)  female  47.0     1     0
2            Myles, Mr. Thomas Francis    male  62.0     0     0
3            Wirz, Mr. Albert    male  27.0     0     0
4 Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0     1     1
..           ...
413           Spector, Mr. Woolf    male   NaN     0     0
414           Oliva y Ocana, Dona. Fermina  female  39.0     0     0
415           Saether, Mr. Simon Sivertsen    male  38.5     0     0
416           Ware, Mr. Frederick    male   NaN     0     0
417           Peter, Master. Michael J    male   NaN     1     1

Ticket      Fare Cabin Embarked
0          330911  7.8292   NaN      Q
1          363272  7.0000   NaN      S
2          240276  9.6875   NaN      Q
3          315154  8.6625   NaN      S
4          3101298 12.2875   NaN      S
..           ...
413          A.5. 3236  8.0500   NaN      S
414          PC 17758 108.9000  C105      C
415  SOTON/O.Q. 3101262  7.2500   NaN      S
416          359309  8.0500   NaN      S
417          2668   22.3583   NaN      C

[418 rows x 12 columns]>
```

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

```
Out[19]: PassengerId      0
Survived        0
Pclass          0
Name            0
Sex             0
Age            86
SibSp          0
Parch          0
Ticket         0
Fare           1
Cabin          327
Embarked        0
dtype: int64
```

```
In [20]: df.drop("Cabin", axis=1, inplace=True)
```

```
In [21]: df["Age"].fillna(df["Age"].mean(), inplace=True)
```

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

```
Out[22]: PassengerId      0
Survived        0
Pclass          0
Name            0
Sex             0
Age             0
SibSp           0
Parch           0
Ticket          0
Fare            1
Embarked        0
dtype: int64
```

```
In [23]: df["Fare"].fillna(df["Fare"].mean(), inplace=True)
```

```
In [24]: df.isnull().sum()
```

```
Out[24]: PassengerId      0
Survived        0
Pclass          0
Name            0
Sex             0
Age             0
SibSp           0
Parch           0
Ticket          0
Fare            0
Embarked        0
dtype: int64
```

```
In [25]: print(df.columns.values
    )
```

```
['PassengerId' 'Survived' 'Pclass' 'Name' 'Sex' 'Age' 'SibSp' 'Parch'
'Ticket' 'Fare' 'Embarked']
```

```
In [26]: df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch
count	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000
mean	30.423610	0.333333	2.333333	30.803514	0.433333	0.383333
std	10.470859	0.473689	0.841838	12.042615	0.896760	0.981429
min	1.000000	0.000000	1.000000	0.130000	0.000000	0.000000
25%	24.500000	0.000000	1.000000	22.000000	0.000000	0.000000
50%	30.423610	0.000000	2.000000	30.803514	0.000000	0.000000
75%	34.175000	1.000000	3.000000	34.600000	1.000000	0.000000
max	80.000000	1.000000	3.000000	80.000000	8.000000	512.320000

In [27]: `df.describe(include=['O'])`

Out[27]:

	Name	Sex	Ticket	Embarked
count	418	418	418	418
unique	418	2	363	3
top	Kelly, Mr. James	male	PC 17608	S
freq	1	266	5	270

In [28]: `df[['Pclass', 'Survived']].groupby(['Pclass'], as_index=False).value_counts().sort`

Out[28]:

Pclass	Survived	count
1	1	50
3	2	30
5	3	72
0	1	57
2	2	63
4	3	146

In [29]: `df[['Pclass', 'Survived']].groupby(['Pclass'], as_index=False).mean().sort_values`

Out[29]:

Pclass	Survived
0	1 0.467290
2	3 0.330275
1	2 0.322581

In [30]: `df[['Sex', 'Survived']].groupby(['Sex'], as_index=False).value_counts().sort_value`

Out[30]:

Sex	Survived	count
0 female	1	152
1 male	0	266

In [31]: `df[['Sex', 'Survived']].groupby(['Sex'], as_index=False).mean().sort_values(by='Su`

Out[31]:

Sex	Survived
0 female	1.0
1 male	0.0

In [32]: `df[['SibSp', 'Survived']].groupby(['SibSp'], as_index=False).value_counts().sort`

Out[32]:

	SibSp	Survived	count
1	0	1	88
3	1	1	54
5	2	1	6
7	3	1	1
9	4	1	1
10	5	1	1
12	8	1	1
0	0	0	195
2	1	0	56
4	2	0	8
6	3	0	3
8	4	0	3
11	8	0	1

In [33]: df[['SibSp', 'Survived']].groupby(['SibSp'], as_index=False).mean().sort_values(by

Out[33]:

	SibSp	Survived
5	5	1.000000
6	8	0.500000
1	1	0.490909
2	2	0.428571
0	0	0.310954
3	3	0.250000
4	4	0.250000

In [34]: df[['Parch', 'Survived']].groupby(['Parch'], as_index=False).value_counts().sort_v

Out[34]:

	Parch	Survived	count
1	0	1	99
2	1	1	28
4	2	1	20
6	3	1	2
8	4	1	2
12	9	1	1
0	0	0	225
3	1	0	24
5	2	0	13
7	3	0	1
9	5	0	1
10	6	0	1
11	9	0	1

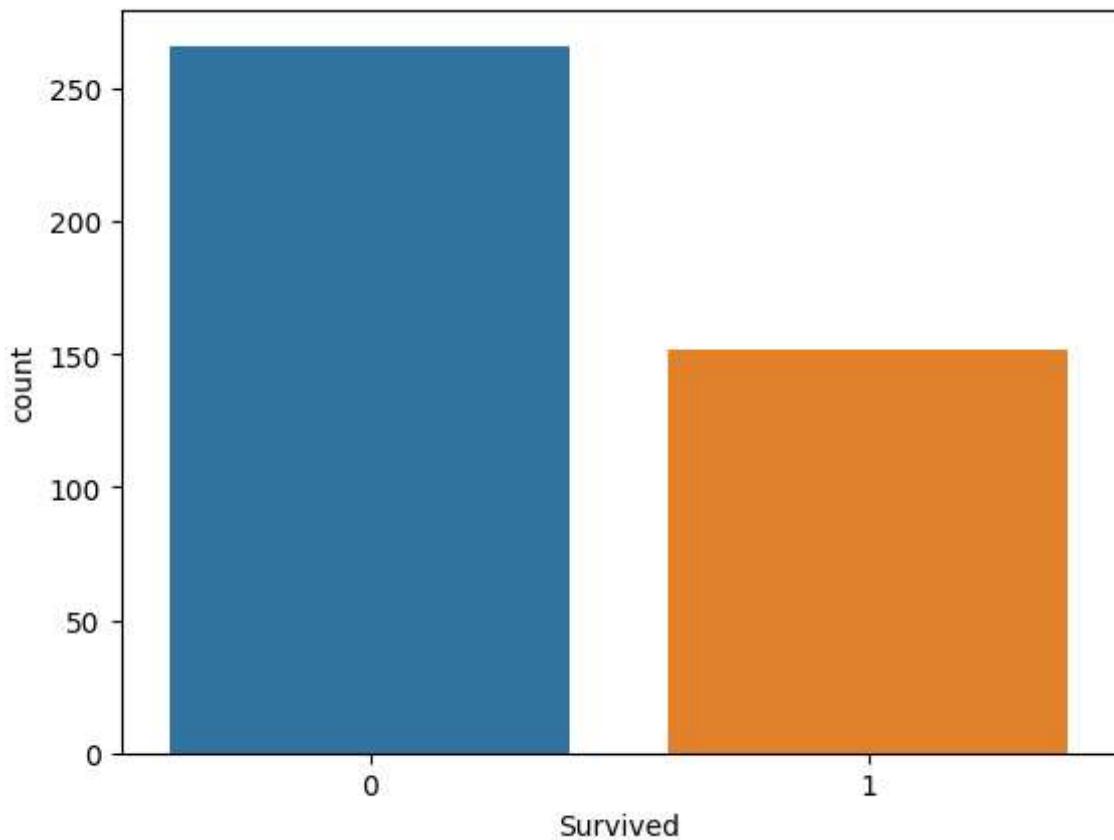
In [35]: `df[['Parch', 'Survived']].groupby(['Parch'], as_index=False).mean().sort_values(by`

Out[35]:

	Parch	Survived
4	4	1.000000
3	3	0.666667
2	2	0.606061
1	1	0.538462
7	9	0.500000
0	0	0.305556
5	5	0.000000
6	6	0.000000

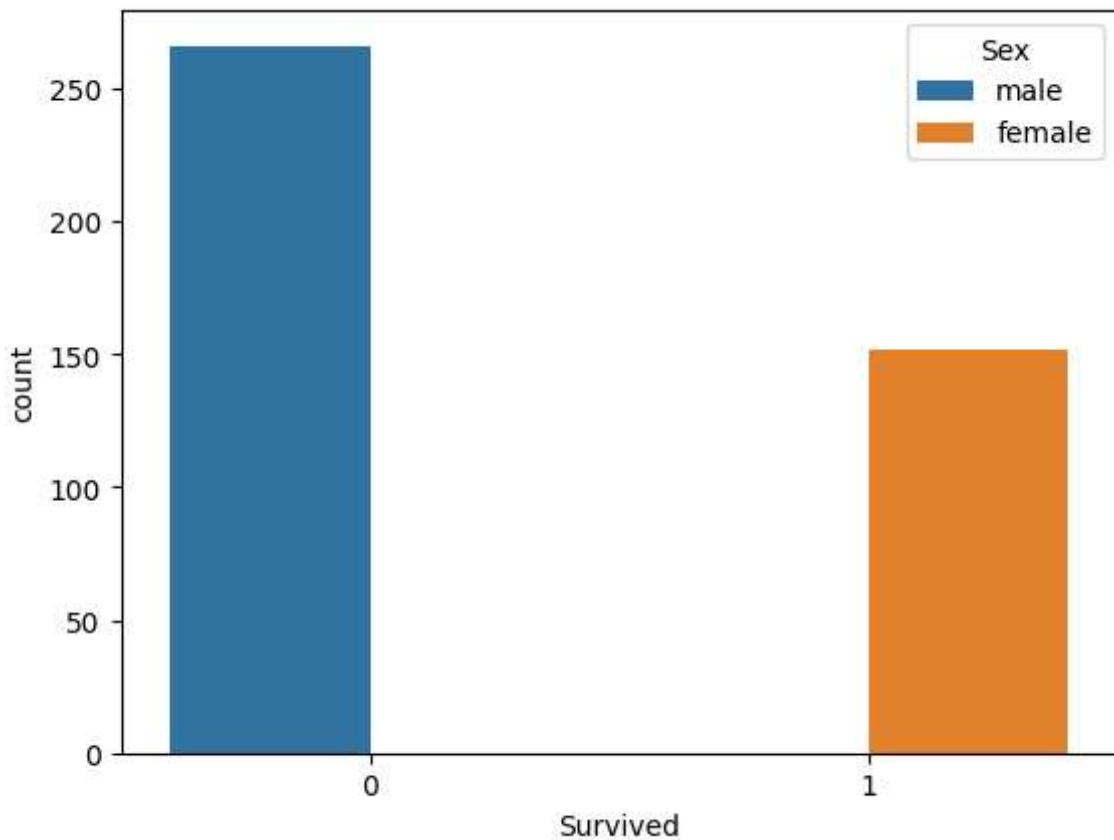
In [36]: `sns.countplot(x='Survived', data=df)`

Out[36]: <Axes: xlabel='Survived', ylabel='count'>



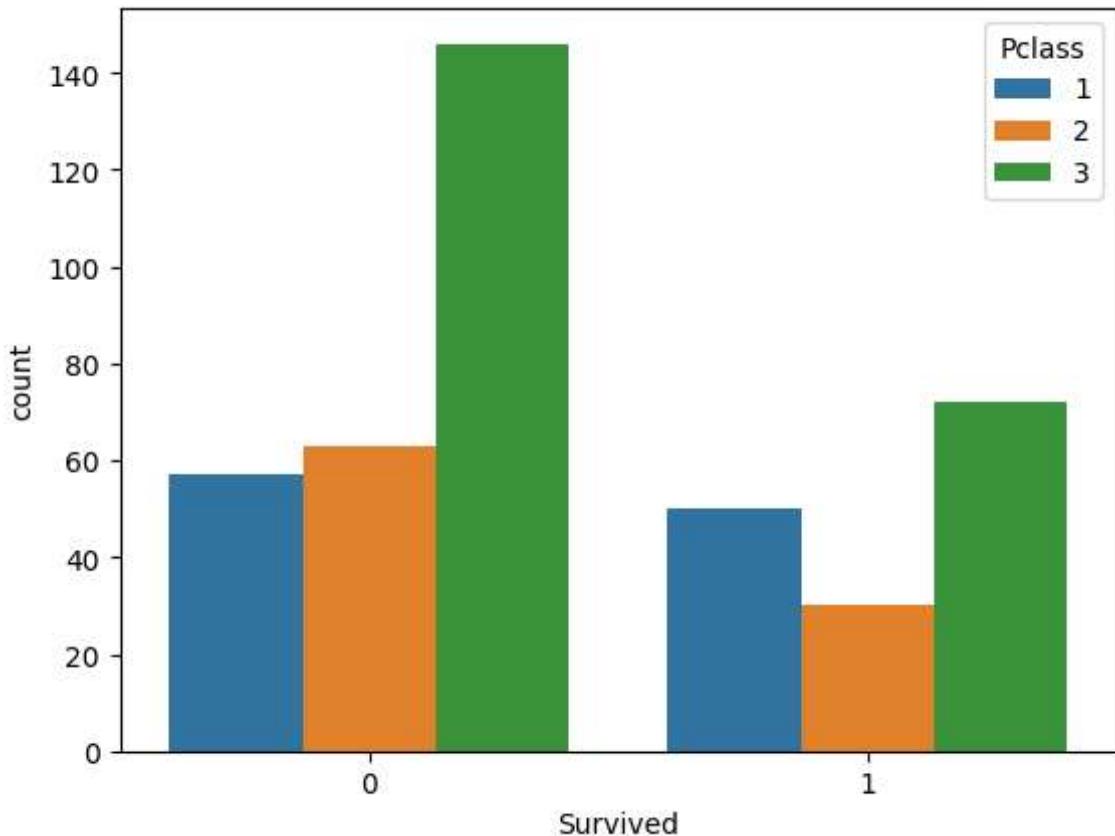
```
In [37]: sns.countplot(x='Survived',hue='Sex',data=df)
```

```
Out[37]: <Axes: xlabel='Survived', ylabel='count'>
```



```
In [38]: sns.countplot(x='Survived',hue='Pclass',data=df)
```

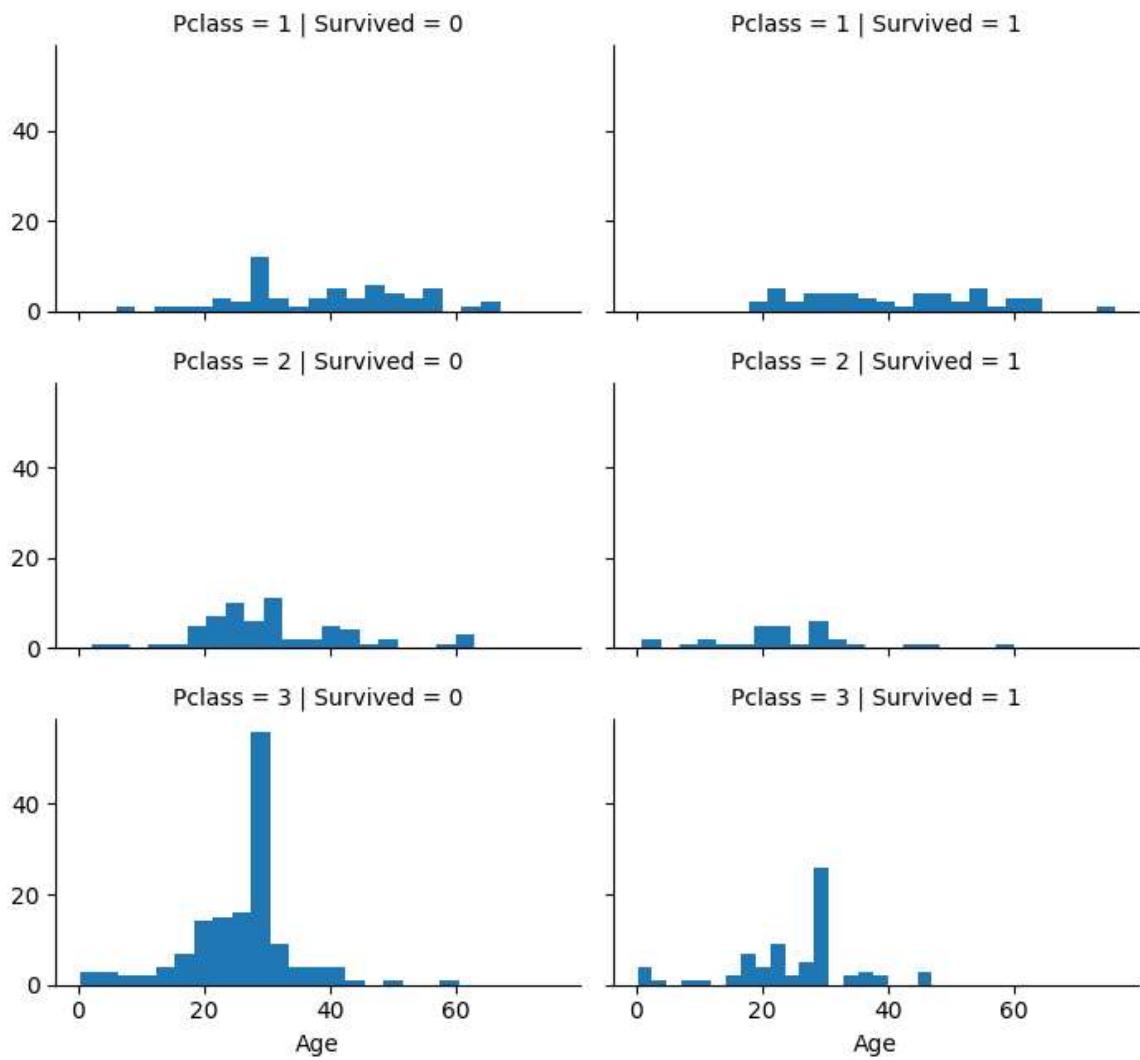
```
Out[38]: <Axes: xlabel='Survived', ylabel='count'>
```



```
In [39]: grid=sns.FacetGrid(df,col='Survived',row='Pclass',height=2.2,aspect=1.6)
grid.map(plt.hist,'Age',alpha=1,bins=20)
grid.add_legend()
```

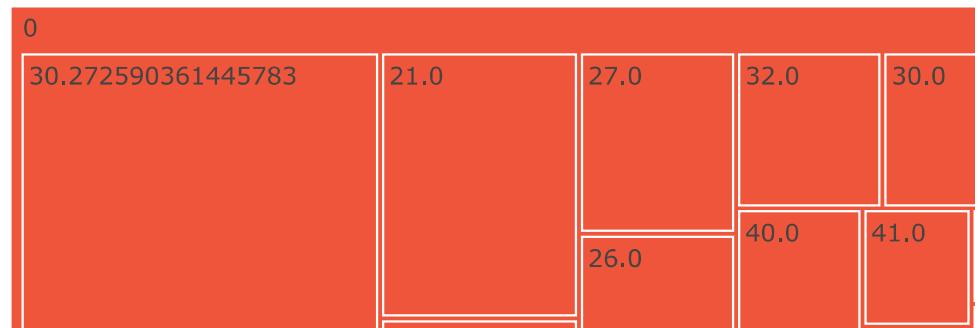
```
C:\Users\Shobhit Khaiwal\AppData\Local\Programs\Python\Python311\Lib\site-packages
s\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
      self._figure.tight_layout(*args, **kwargs)
```

```
Out[39]: <seaborn.axisgrid.FacetGrid at 0x28373123c10>
```



```
In [40]: import plotly.express as px
```

```
In [41]: px.treemap(df, path=['Survived', 'Age'], values='Pclass', color='Sex',  
                 hover_data=['Survived'], color_continuous_scale='RdBu')
```



```
In [42]: df['Embarked'].value_counts()
```

```
Out[42]: Embarked
S    270
C    102
Q     46
Name: count, dtype: int64
```

```
In [43]: df.replace({'Sex':{'male':1,'female':0}, 'Embarked':{'S':0,'C':1,'Q':2}},inplace=True)
```

```
In [44]: df.head()
```

Out[44]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	892	0	3	Kelly, Mr. James	1	34.5	0	0	330911	7.8292
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	0	47.0	1	0	363272	7.0000
2	894	0	2	Myles, Mr. Thomas Francis	1	62.0	0	0	240276	9.6875
3	895	0	3	Wirz, Mr. Albert	1	27.0	0	0	315154	8.6625
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	22.0	1	1	3101298	12.2875

In [45]: `df.tail()`

Out[45]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
413	1305	0	3	Spector, Mr. Woolf	1	30.27259	0	0	A.5. 3236
414	1306	1	1	Oliva y Ocana, Dona. Fermina	0	39.00000	0	0	PC 17758
415	1307	0	3	Saether, Mr. Simon Sivertsen	1	38.50000	0	0	SOTON/O.Q. 3101262
416	1308	0	3	Ware, Mr. Frederick	1	30.27259	0	0	359309
417	1309	0	3	Peter, Master. Michael J	1	30.27259	1	1	2668

In [46]: `A=df.drop(columns=['Name','Ticket','PassengerId','Survived'],axis=1)`

In [47]: `B=df['Survived']`

In [48]: `A`

Out[48]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	34.50000	0	0	7.8292	2
1	3	0	47.00000	1	0	7.0000	0
2	2	1	62.00000	0	0	9.6875	2
3	3	1	27.00000	0	0	8.6625	0
4	3	0	22.00000	1	1	12.2875	0
...
413	3	1	30.27259	0	0	8.0500	0
414	1	0	39.00000	0	0	108.9000	1
415	3	1	38.50000	0	0	7.2500	0
416	3	1	30.27259	0	0	8.0500	0
417	3	1	30.27259	1	1	22.3583	1

418 rows × 7 columns

In [49]:

B

Out[49]:

```
0      0
1      1
2      0
3      0
4      1
 ..
413    0
414    1
415    0
416    0
417    0
```

Name: Survived, Length: 418, dtype: int64

In [51]:

`from sklearn.model_selection import train_test_split as tts`

In [52]:

`A_train,A_test,B_train,B_test=tts(A,B,test_size=0.2,random_state=2)`

In [53]:

`print(A_train.shape,A.shape,A_test.shape)`

(334, 7) (418, 7) (84, 7)

In [54]:

`from sklearn.linear_model import LogisticRegression as lg`

In [55]:

`model=lg()`

In [57]:

`model.fit(A_train,B_train)`

```
C:\Users\Shobhit Khaiwal\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning:  
lbfgs failed to converge (status=1):  
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.  
  
Increase the number of iterations (max_iter) or scale the data as shown in:  
    https://scikit-learn.org/stable/modules/preprocessing.html  
Please also refer to the documentation for alternative solver options:  
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
```

Out[57]:

```
▼ LogisticRegression  
LogisticRegression()
```

In [58]:

```
A_train_prediction=model.predict(A_train)
```

In [60]:

```
print(A_train_prediction  
)
```

```
[1 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 1 0 1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0  
1 1 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 0 1  
0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 0  
1 1 0 0 0 0 1 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 1 0 0  
0 0 1 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 0 1 0 0 0 0 1 0 1 1  
1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 0 1 0 0 1 0 0  
1 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1  
0 1 1 1 1 0 0 0 1 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0  
0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0  
1]
```

In [62]:

```
from sklearn.metrics import accuracy_score as acs
```

In [64]:

```
training_data_accuracy=acs(B_train,A_train_prediction)
```

In [65]:

```
print('accuracy_Score_of_trainning_data :',training_data_accuracy)
```

```
accuracy_Score_of_trainning_data : 1.0
```

In [67]:

```
A_test_prediction=model.predict(A_test)
```

In [68]:

```
print(A_test_prediction)
```

```
[0 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 1 0 0 1  
1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0  
0 1 1 0 1 0 0 0 0 0]
```

In [69]:

```
test_data_accuracy=acs(B_test,A_test_prediction)  
print('accuracy_Score_of_test_data :',test_data_accuracy)
```

```
accuracy_Score_of_test_data : 1.0
```

In []: