Exploratory Data Analysis (EDA)

Steps in Data Analysis

1. Data collection
2. EDA
3. preprocessing
4. Model Building
5. Evalute the Model

In [2]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
df=pd.read_csv("train.csv")
```

In [116…
```
df.head()
```

Out[116]:

	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	Heikkinen, 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

1. Size of the data

In [5]:
```
df.shape
```

Out[5]:
```
(891, 12)
```

In [6]:
```
df.size
```

Out[6]:
```
10692
```

In [7]:
```
df.memory_usage()
```

Out[7]:
```
Index           128
PassengerId    7128
Survived       7128
Pclass         7128
Name           7128
Sex            7128
Age            7128
SibSp          7128
Parch          7128
Ticket         7128
Fare           7128
Cabin          7128
Embarked       7128
dtype: int64
```

In [8]:
```
df.memory_usage(deep=True)
```

Out[8]:
```
Index           128
PassengerId    7128
Survived       7128
Pclass         7128
Name          74813
Sex           54979
Age            7128
SibSp          7128
Parch          7128
Ticket        56802
Fare           7128
Cabin         34344
Embarked      51626
dtype: int64
```

2. How data looks like

In [9]:
```
df.head()
```

Out[9]:

	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	Heikkinen, 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

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

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [11]:
```
df.sample()
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
623	624	0	3	Hansen, Mr. Henry Damsgaard	male	21.0	0	0	350029	7.8542	NaN	S

In [12]:
```
df.sample(5)
```

Out[12]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
12	13	0	3	Saundercock, Mr. William Henry	male	20.0	0	0	A/5. 2151	8.0500	NaN	S
625	626	0	1	Sutton, Mr. Frederick	male	61.0	0	0	36963	32.3208	D50	S
449	450	1	1	Peuchen, Major. Arthur Godfrey	male	52.0	0	0	113786	30.5000	C104	S
675	676	0	3	Edvardsson, Mr. Gustaf Hjalmar	male	18.0	0	0	349912	7.7750	NaN	S
444	445	1	3	Johannesen-Bratthammer, Mr. Bernt	male	NaN	0	0	65306	8.1125	NaN	S

3. what is data type of column

In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

4.Data Looks like methematicaly

In [14]:
```
df.describe()
```

Out[14]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [15]:
```
df.describe().T
```

Out[15]:

	count	mean	std	min	25%	50%	75%	max
PassengerId	891.0	446.000000	257.353842	1.00	223.5000	446.0000	668.5	891.0000
Survived	891.0	0.383838	0.486592	0.00	0.0000	0.0000	1.0	1.0000
Pclass	891.0	2.308642	0.836071	1.00	2.0000	3.0000	3.0	3.0000
Age	714.0	29.699118	14.526497	0.42	20.1250	28.0000	38.0	80.0000
SibSp	891.0	0.523008	1.102743	0.00	0.0000	0.0000	1.0	8.0000
Parch	891.0	0.381594	0.806057	0.00	0.0000	0.0000	0.0	6.0000
Fare	891.0	32.204208	49.693429	0.00	7.9104	14.4542	31.0	512.3292

5. check missing value

In [16]:
```
df.isnull()
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
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 [18]:
```
df.isnull().sum().sum()
```

Out[18]:
```
866
```

6. Duplicate values

In [19]:
```
df.duplicated()
```

Out[19]:
```
0      False
1      False
2      False
3      False
4      False
       ...
886    False
887    False
888    False
889    False
890    False
Length: 891, dtype: bool
```

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

Out[20]:
```
0
```

In [21]:
```
df[df.duplicated()]
```

Out[21]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

7.Checks unique values

In [22]:
```
df.nunique()
```

Out[22]:
```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

8. Correlations

In [23]:
```
data_corr1=df.corr()
```

In [24]:
```
data_corr1
```

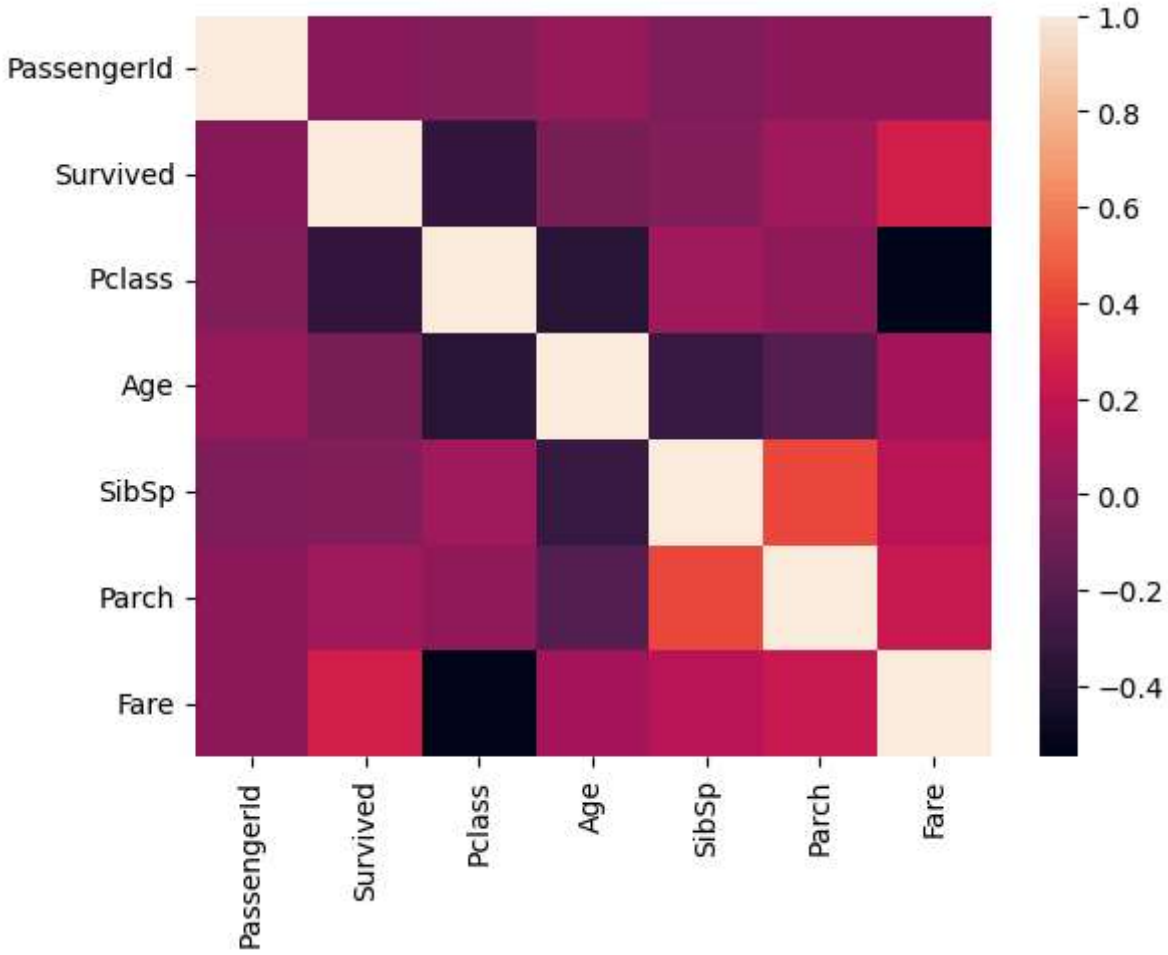
Out[24]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

9. HeatMap

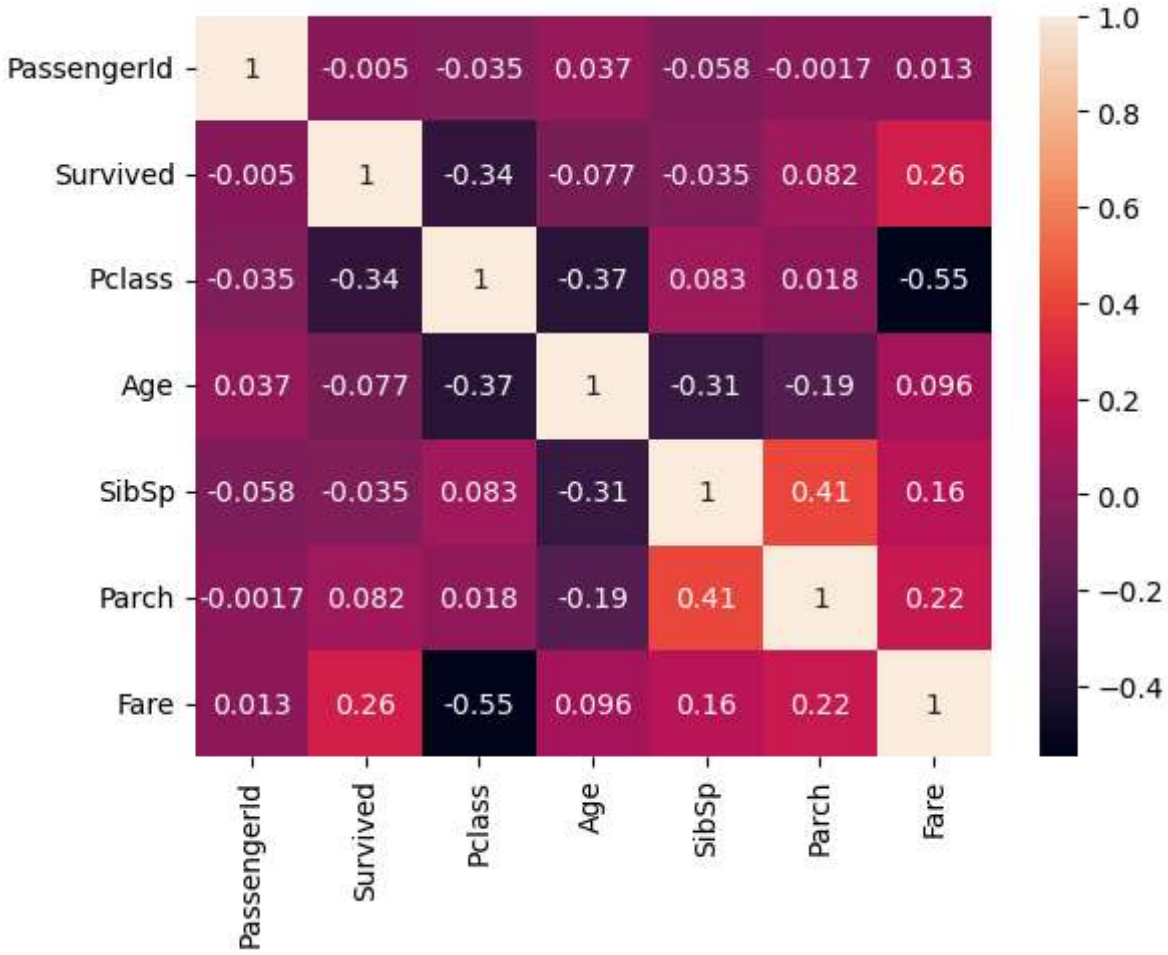
In [25]:
```
sns.heatmap(data_corr1)
```

Out[25]:
```
<AxesSubplot:>
```



In [26]:
```
sns.heatmap(data_corr1,annot=True)
```

Out[26]:
```
<AxesSubplot:>
```



In [27]:
```
df.columns
```

Out[27]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [28]:
```
cat_fea=[column for column in df.columns if df[column].dtype=="O"]
```

In [29]:
```
num_fea=[column for column in df.columns if df[column].dtype!="O"]
```

In [30]:
```
df[cat_fea]
```

Out[30]:

	Name	Sex	Ticket	Cabin	Embarked
0	Braund, Mr. Owen Harris	male	A/5 21171	NaN	S
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	PC 17599	C85	C
2	Heikkinen, Miss. Laina	female	STON/O2. 3101282	NaN	S
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	113803	C123	S
4	Allen, Mr. William Henry	male	373450	NaN	S
...	...	...	...	...	...
886	Montvila, Rev. Juozas	male	211536	NaN	S
887	Graham, Miss. Margaret Edith	female	112053	B42	S
888	Johnston, Miss. Catherine Helen "Carrie"	female	W./C. 6607	NaN	S
889	Behr, Mr. Karl Howell	male	111369	C148	C
890	Dooley, Mr. Patrick	male	370376	NaN	Q

891 rows × 5 columns

In [31]:
```
df[num_fea]
```

Out[31]:

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

891 rows × 7 columns

In [32]:
```
#catagarical varible
#numerical varible
```

Univarite Data Analysis

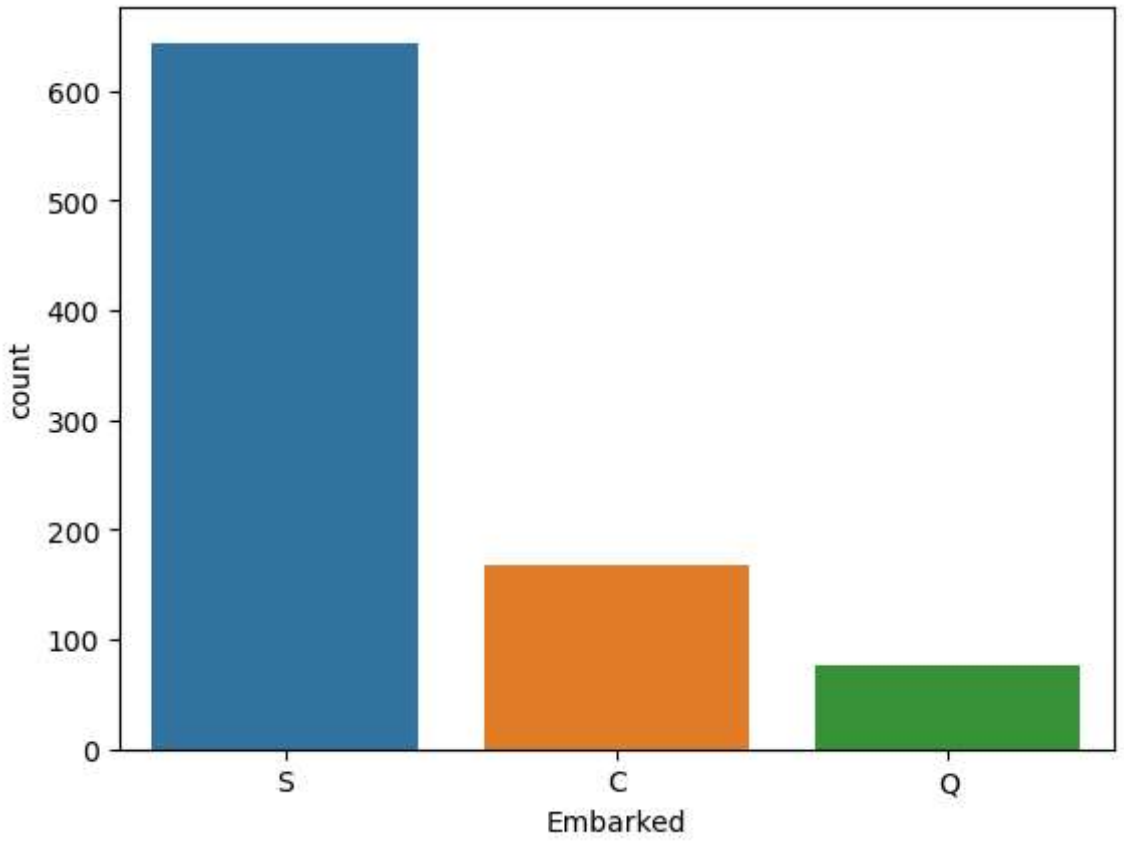
Catagarical Varible

1. Count plot

In [33]:
```
sns.countplot(df["Embarked"])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as
a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

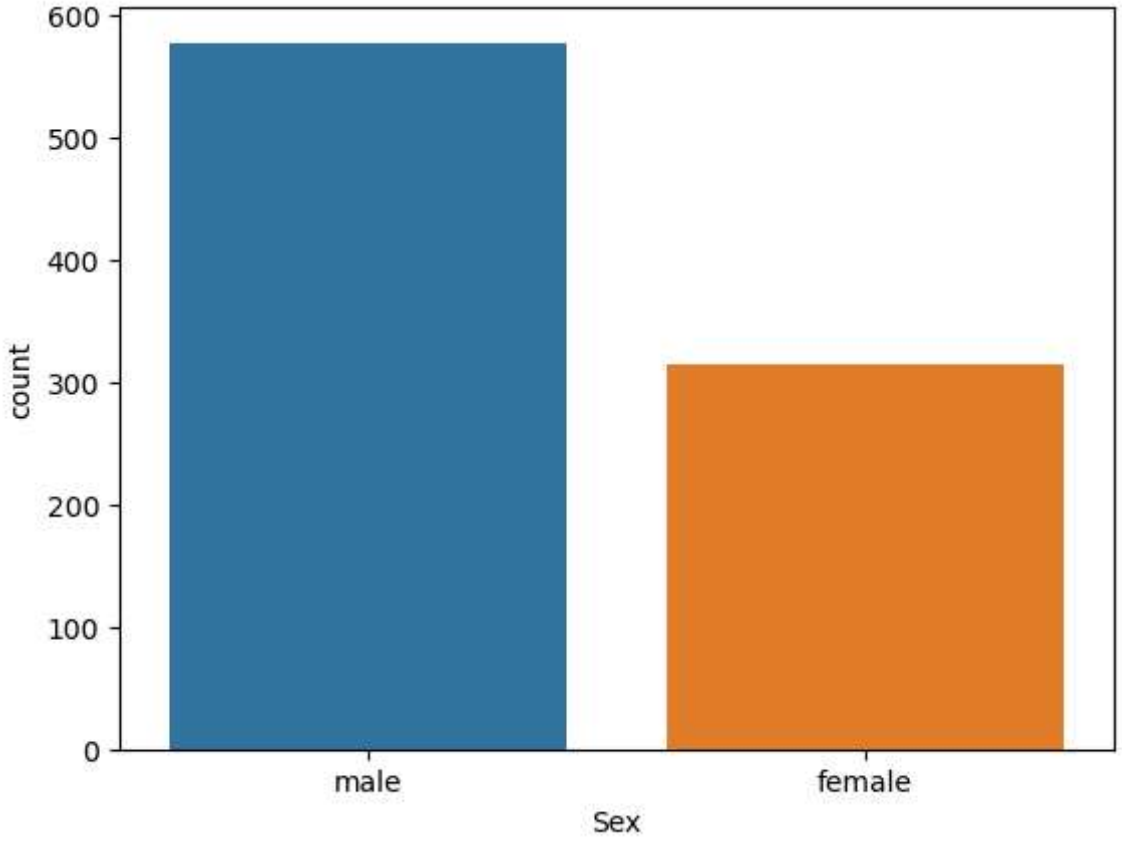
Out[33]: <AxesSubplot:xlabel='Embarked', ylabel='count'>



In [34]:
```
sns.countplot(df["Sex"])
```

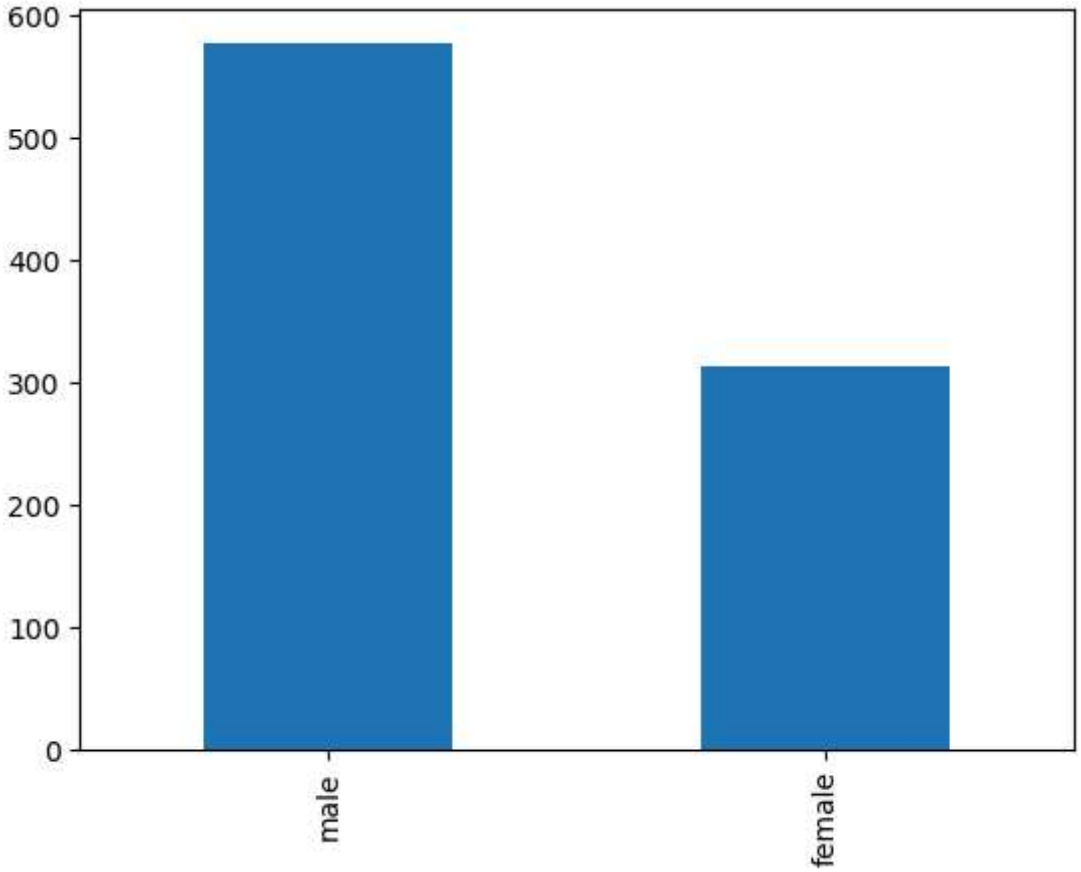
```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as
a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[34]: <AxesSubplot:xlabel='Sex', ylabel='count'>



In [35]:
```
df["Sex"].value_counts().plot(kind="bar")
```

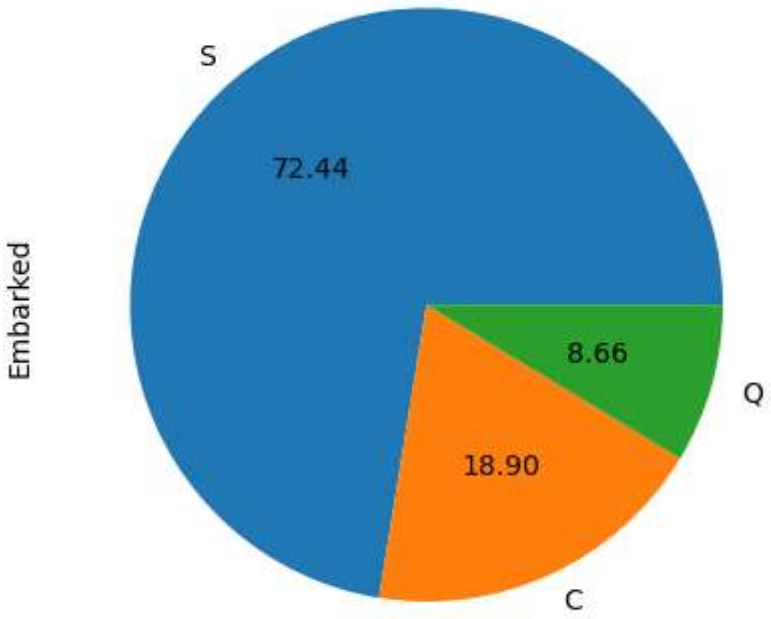
Out[35]:
```
<AxesSubplot:>
```



1. Pie chart

In [36]:
```
df["Embarked"].value_counts().plot(kind="pie",autopct='%.2f')
```

Out[36]:
```
<AxesSubplot:ylabel='Embarked'>
```



Numberical data

In [37]:
```
df[num_fea]
```

Out[37]:

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

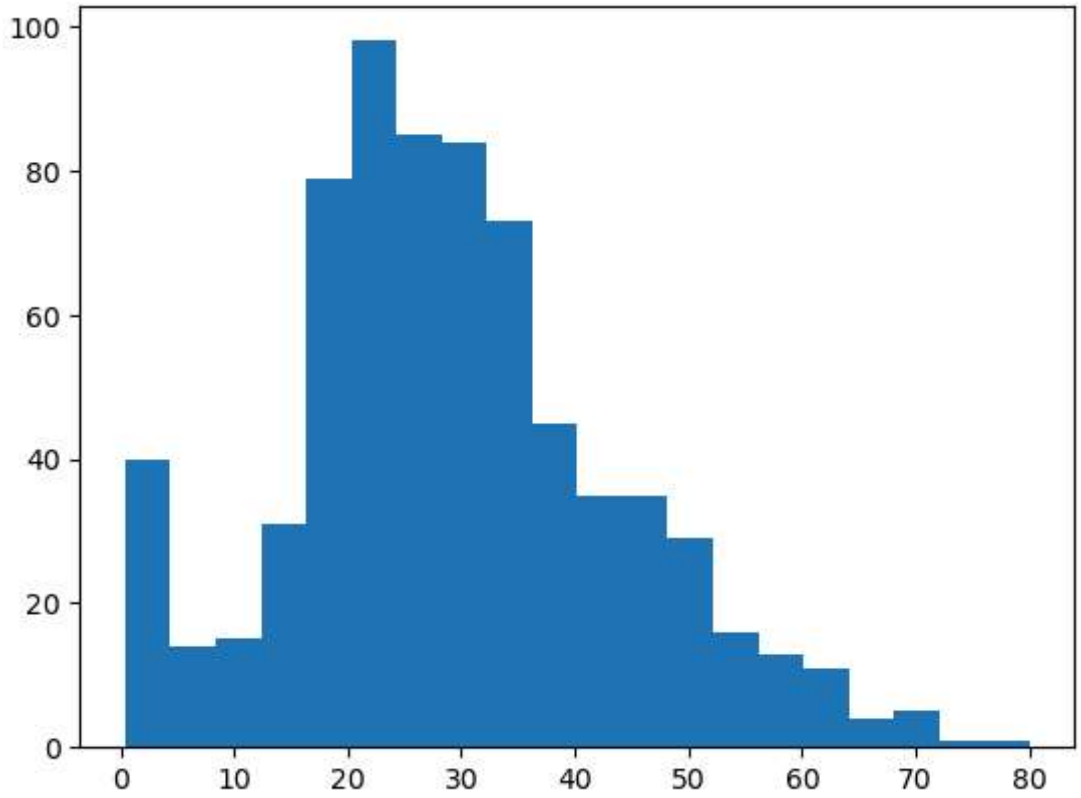
891 rows × 7 columns

Histrogram

In [38]:
```
import matplotlib.pyplot as plt
```

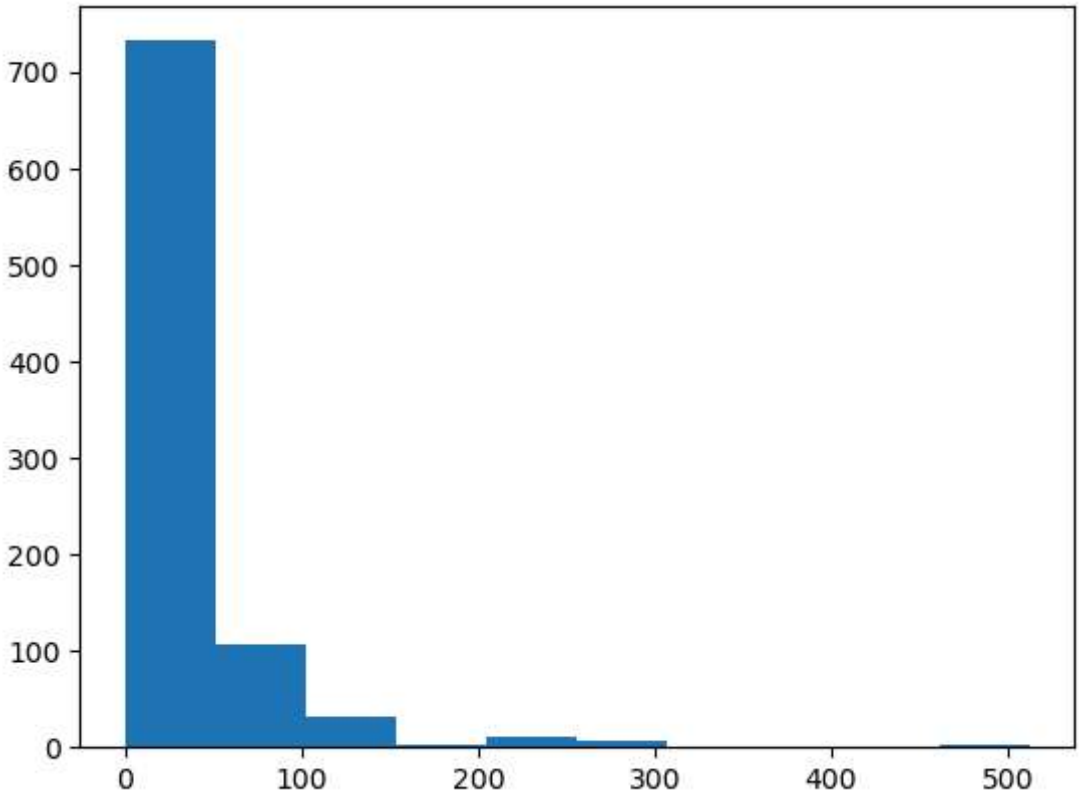
In [39]:
```
plt.hist(df["Age"],bins=20)
```

Out[39]:
```
(array([40., 14., 15., 31., 79., 98., 85., 84., 73., 45., 35., 35., 29.,
        16., 13., 11.,  4.,  5.,  1.,  1.]),
 array([ 0.42 ,  4.399,  8.378, 12.357, 16.336, 20.315, 24.294, 28.273,
        32.252, 36.231, 40.21 , 44.189, 48.168, 52.147, 56.126, 60.105,
        64.084, 68.063, 72.042, 76.021, 80.   ]),
 <BarContainer object of 20 artists>)
```



In [40]:
```
plt.hist(df["Fare"])
```

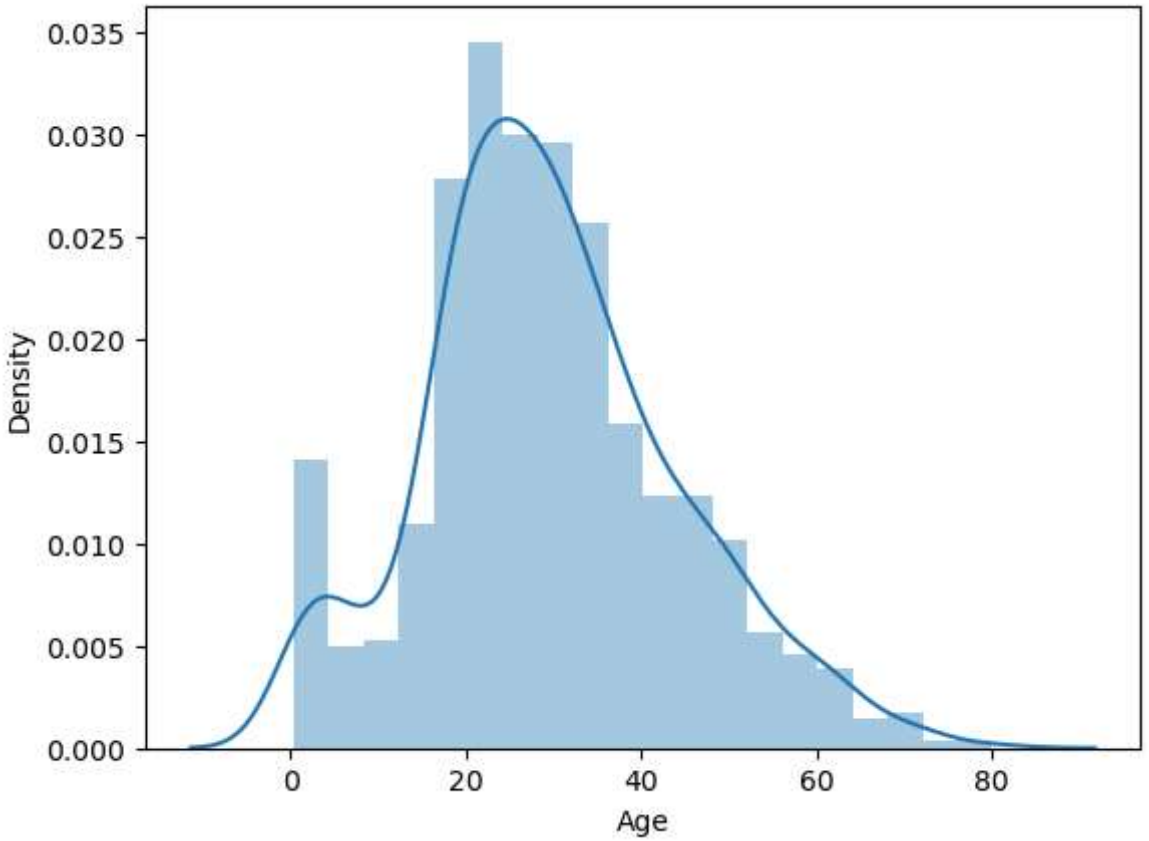
Out[40]:
```
(array([732., 106.,  31.,   2.,  11.,   6.,   0.,   0.,   0.,   3.]),
 array([  0.     ,  51.23292, 102.46584, 153.69876, 204.93168, 256.1646 ,
        307.39752, 358.63044, 409.86336, 461.09628, 512.3292 ]),
 <BarContainer object of 10 artists>)
```



In [41]:
```
sns.distplot(df["Age"])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

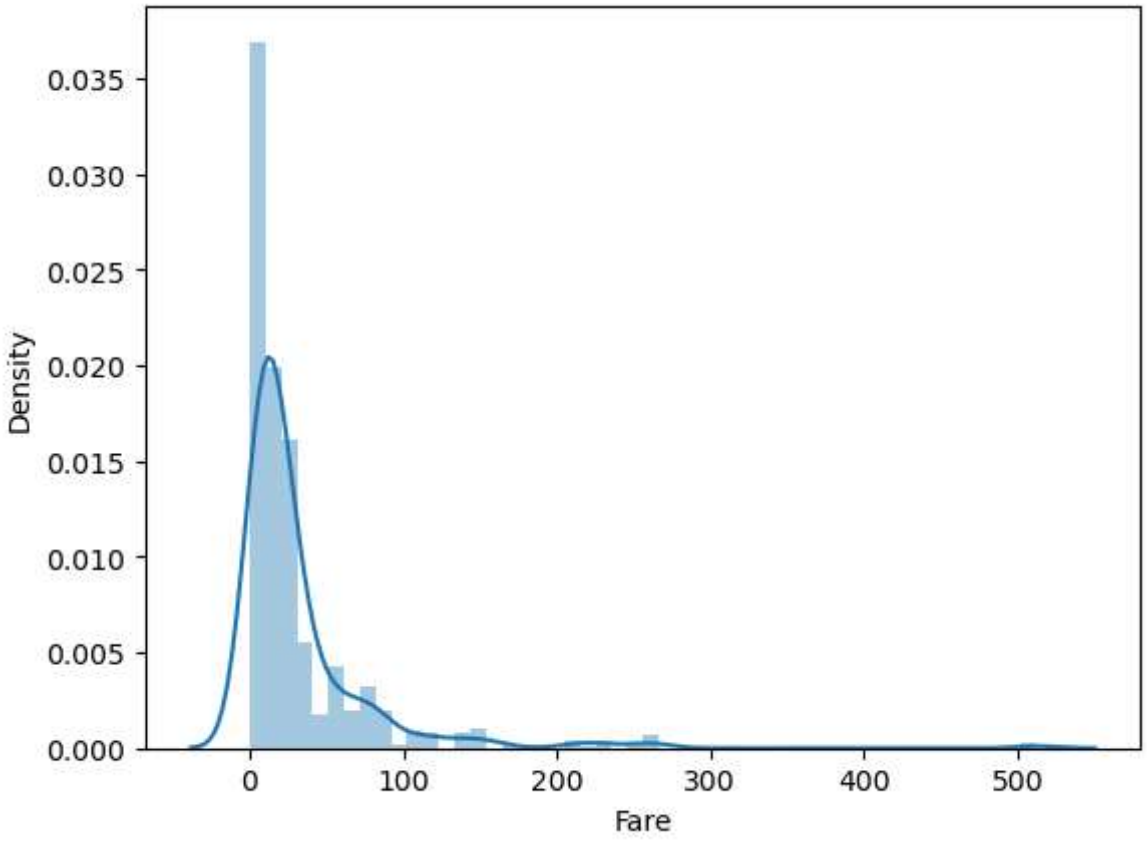
Out[41]:
```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```



In [42]:
```
sns.distplot(df["Fare"])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

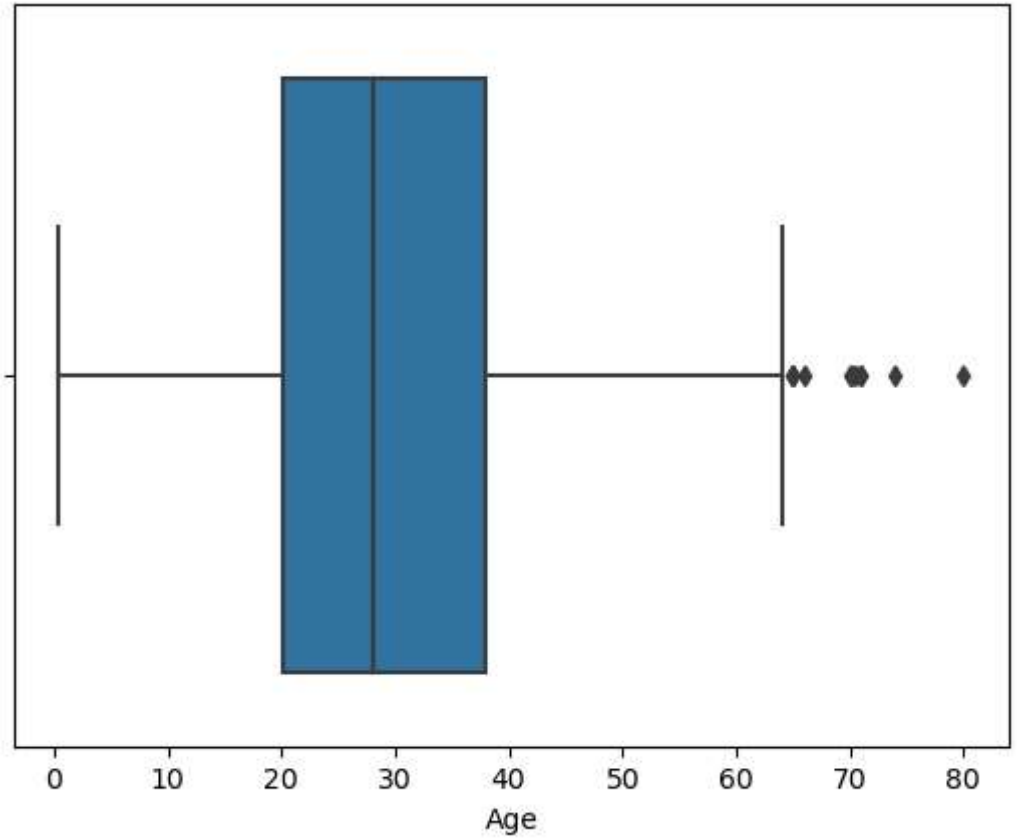
Out[42]:
```
<AxesSubplot:xlabel='Fare', ylabel='Density'>
```



In [43]:
```
sns.boxplot(df['Age'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as
a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[43]:
```
<AxesSubplot:xlabel='Age'>
```



In [44]:
```
df["Age"].min()
```

Out[44]:
```
0.42
```

In [45]:
```
df["Age"].max()
```

Out[45]:
```
80.0
```

In [46]:
```
df["Age"].mean()
```

Out[46]:
```
29.69911764705882
```

In [47]:
```
df["Age"].median()
```

Out[47]:
```
28.0
```

In [48]:
```
df["Age"].skew()*100
```

Out[48]:
```
38.910778230082705
```

In [49]:
```
1-0.38910778230082704
```

Out[49]:
```
0.6108922176991729
```

Bivarinte and Multivarient Data Analysis

catagarical and Numerical obervation are consider in X and Y axies resplitively

In [50]:
```
tips=sns.load_dataset('tips')
```

In [51]:
```
flights=sns.load_dataset('flights')
```

In [52]:
```
iris=sns.load_dataset('iris')
```

In [53]:
```
iris
```

Out[53]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [54]:
```
flights
```

Out[54]:

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121
...	...	...	...
139	1960	Aug	606
140	1960	Sep	508
141	1960	Oct	461
142	1960	Nov	390
143	1960	Dec	432

144 rows × 3 columns

In [55]:
```
tips
```

Out[55]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

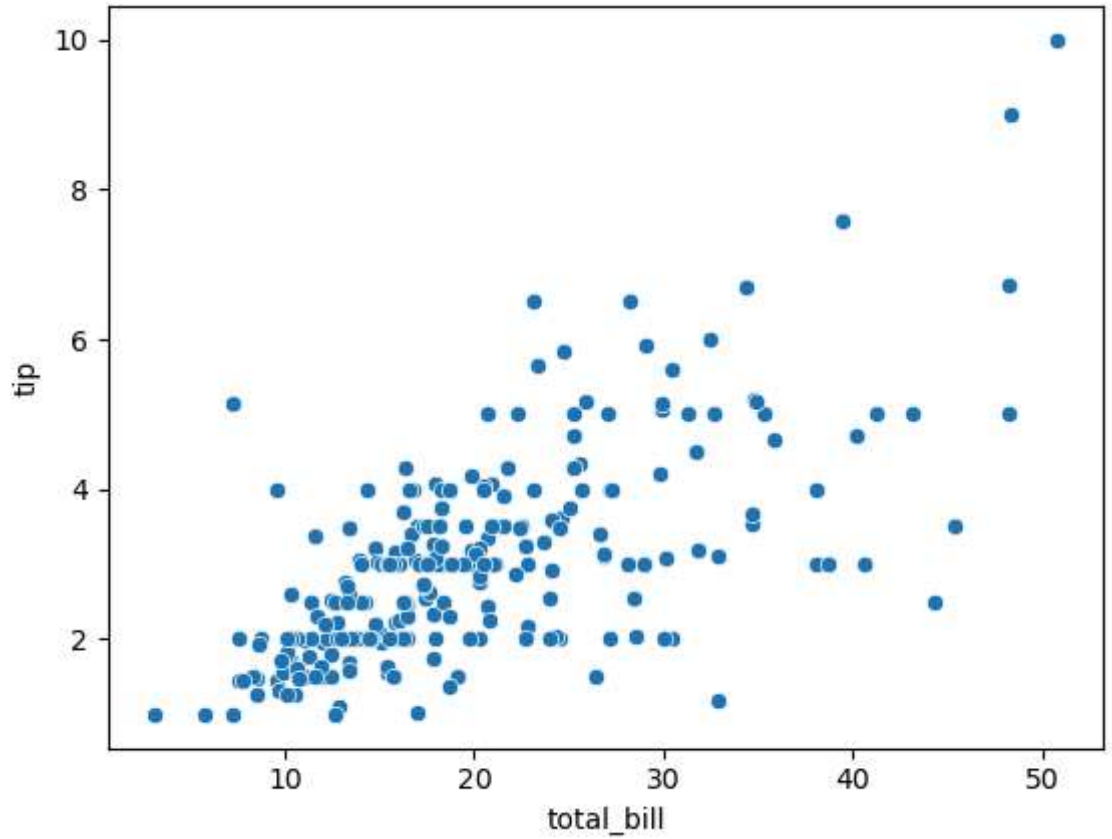
244 rows × 7 columns

scatterplot of multiple varient

In [56]:
```
sns.scatterplot(tips['total_bill'],tips['tip'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

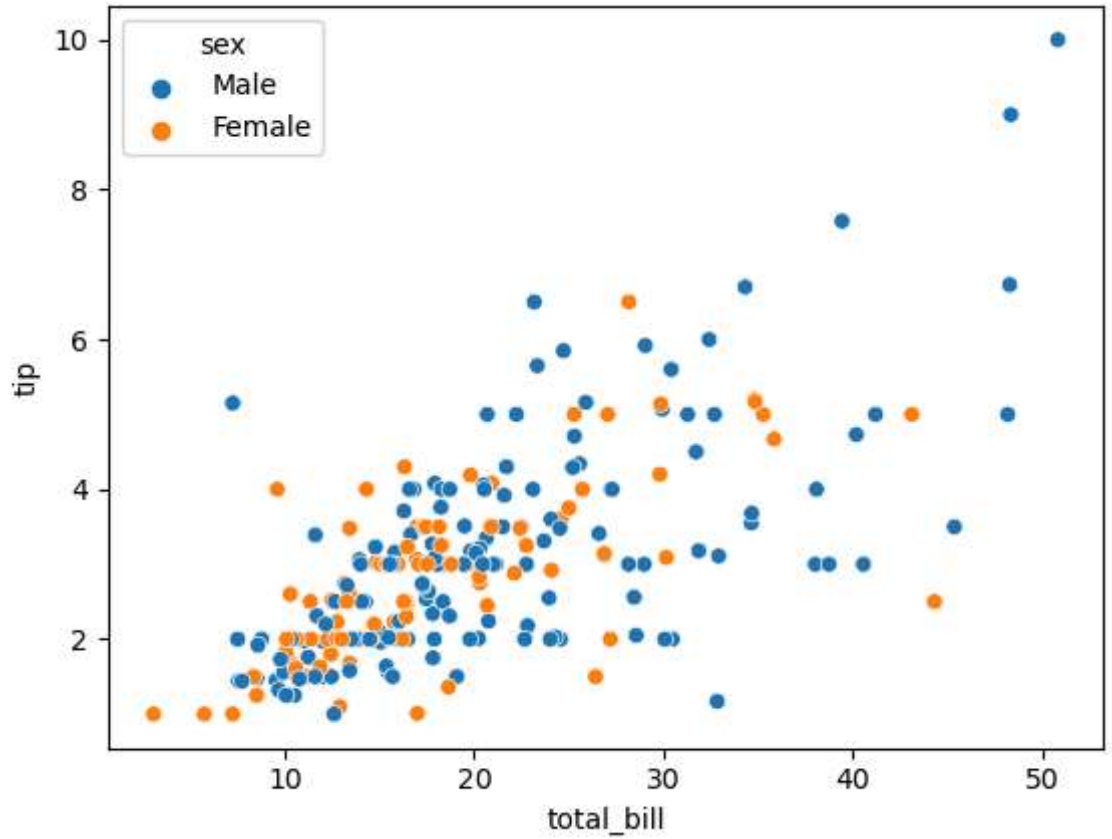
Out[56]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



In [57]:
```
sns.scatterplot(tips['total_bill'],tips['tip'],hue=tips['sex'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

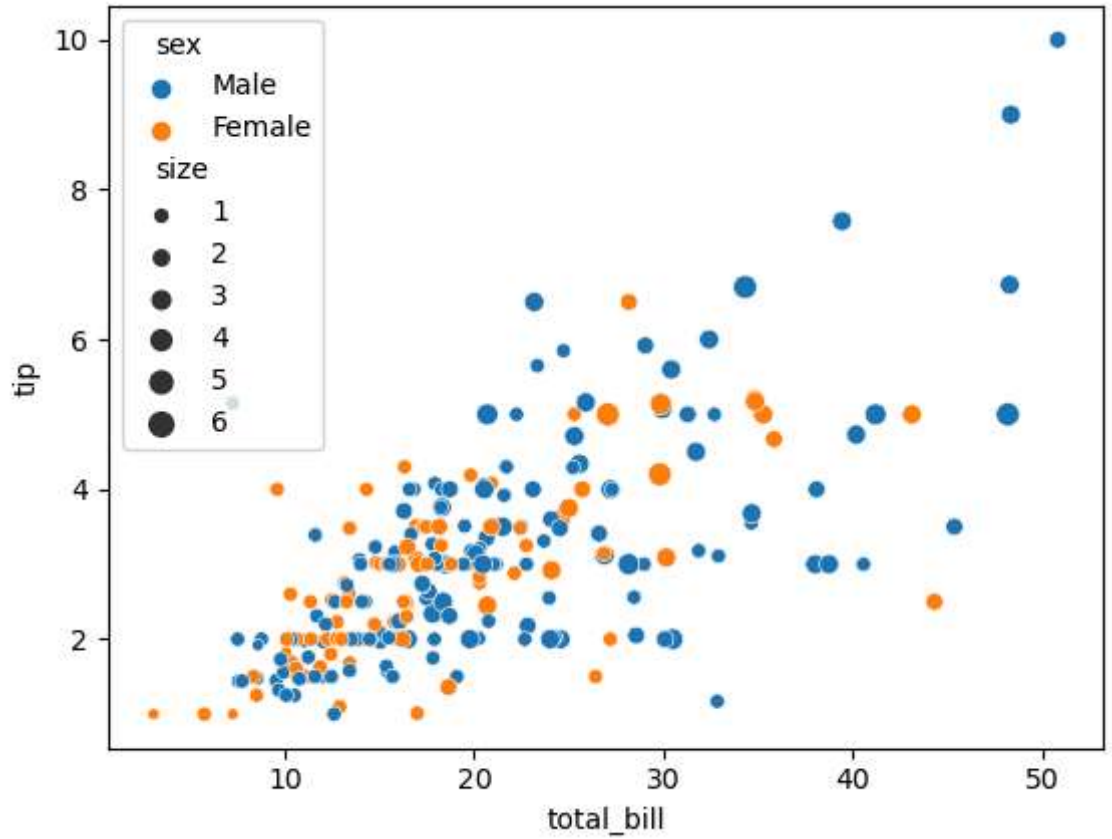
Out[57]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



In [58]:
```
sns.scatterplot(tips['total_bill'],tips['tip'],hue=tips['sex'],size=tips['size'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[58]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```

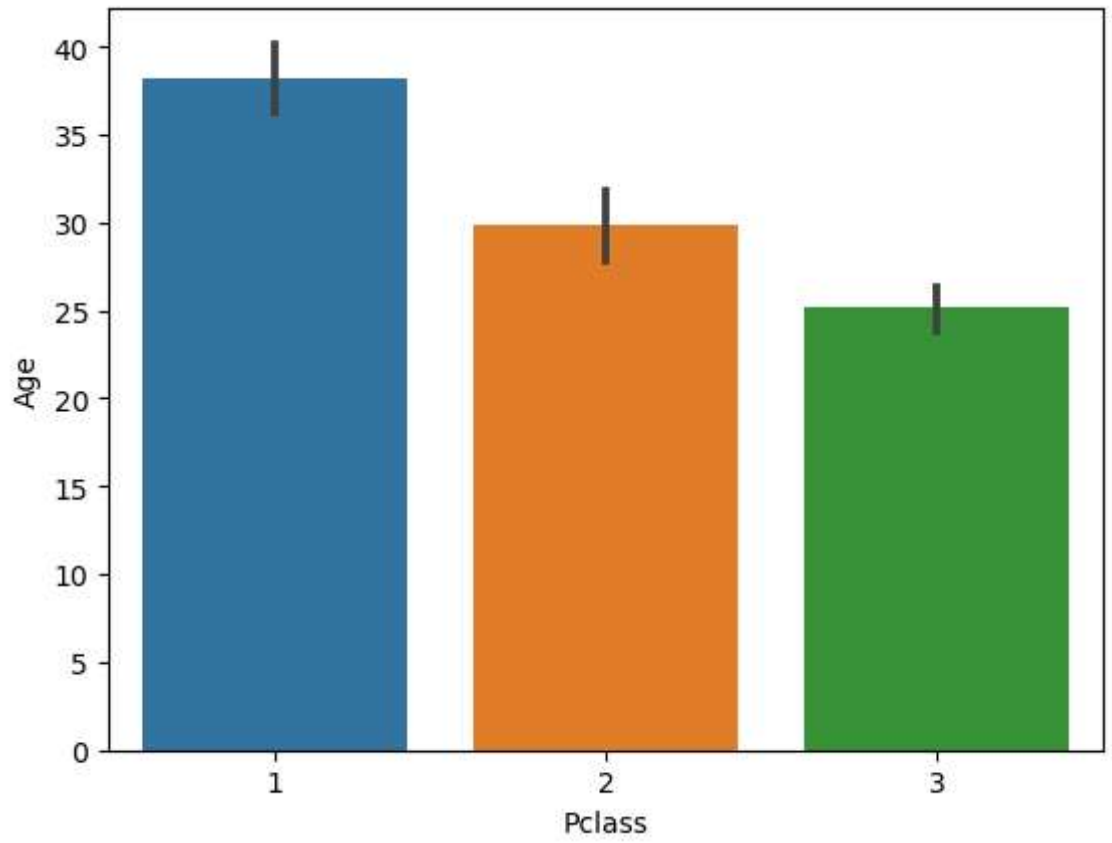


Baraplot on Multiple Varient

In [59]:
```
sns.barplot(df['Pclass'],df['Age'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

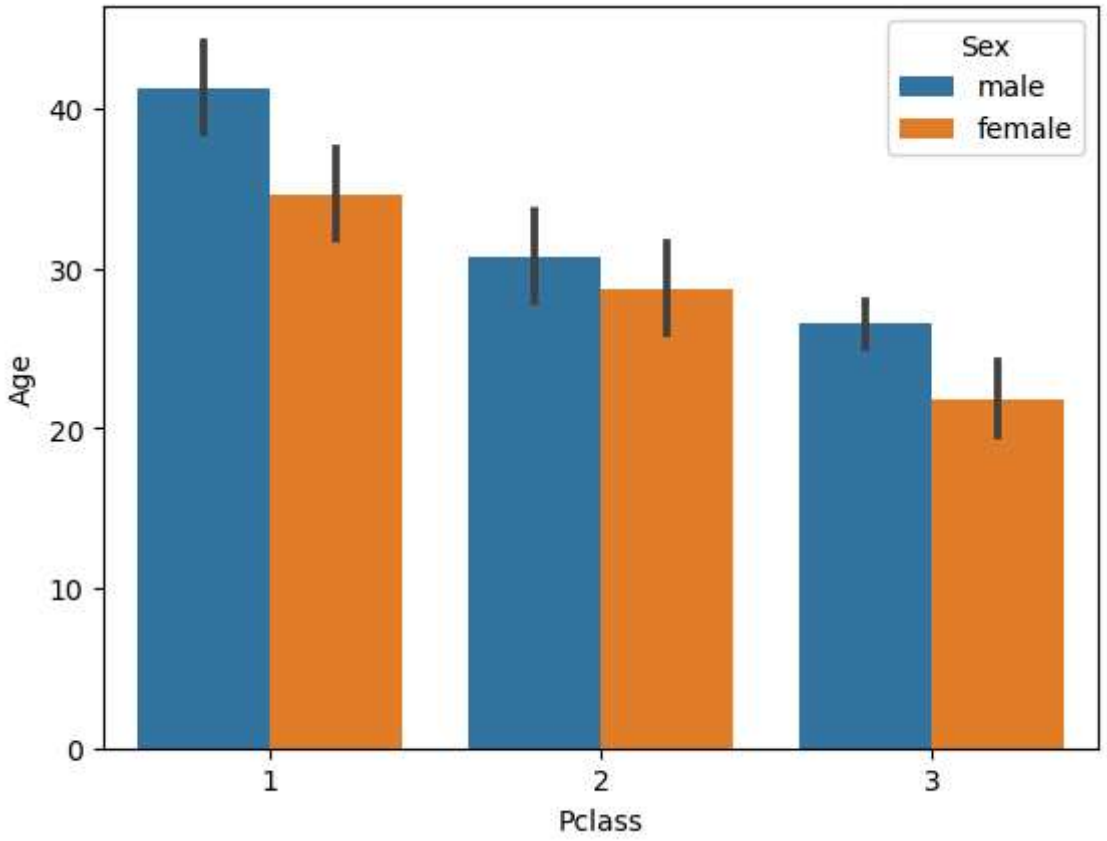
Out[59]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Age'>
```



In [60]:
```
sns.barplot(df['Pclass'],df['Age'],hue=df['Sex'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[60]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Age'>
```

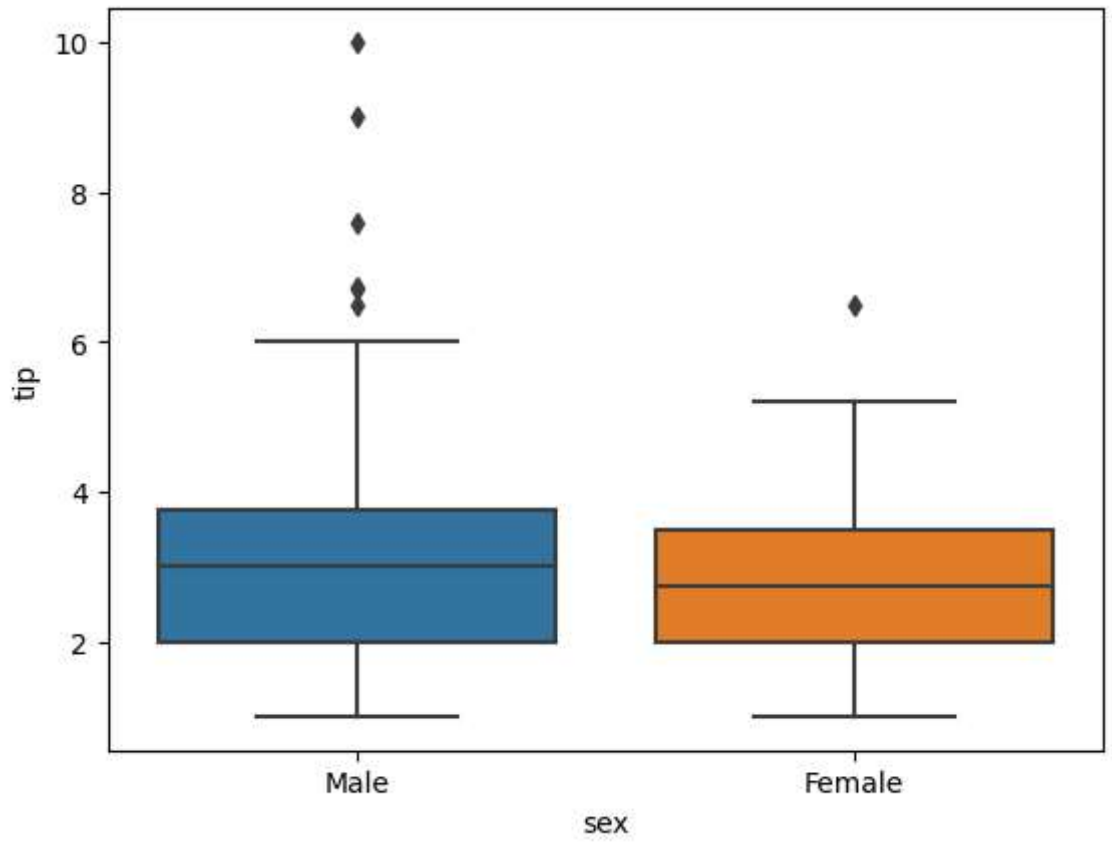


Boxplot with respective multi varient

In [61]:
```
sns.boxplot(tips['sex'],tips['tip'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

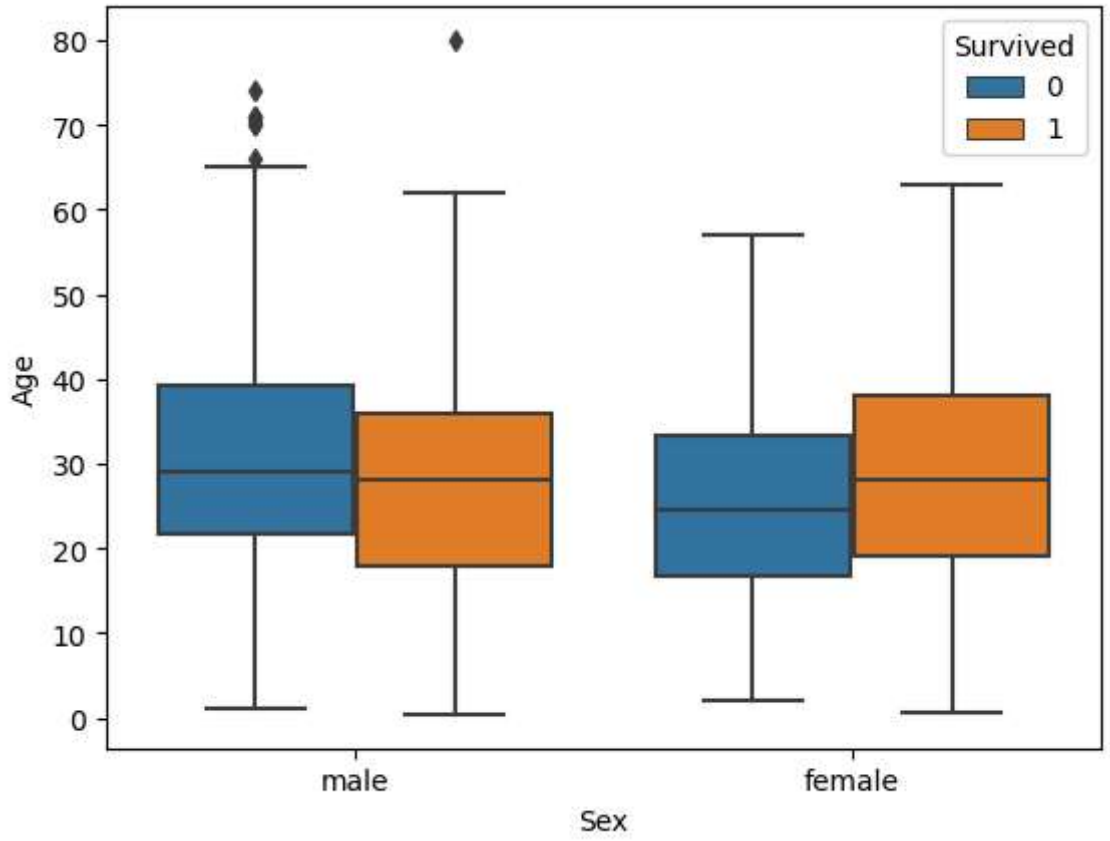
Out[61]:
```
<AxesSubplot:xlabel='sex', ylabel='tip'>
```



In [62]:
```
sns.boxplot(df['Sex'],df['Age'],hue=df['Survived'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[62]:
```
<AxesSubplot:xlabel='Sex', ylabel='Age'>
```



In [63]:
```
df[df['Survived']==0]
```

Out[63]:

	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
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500	NaN	S
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

549 rows × 12 columns

In [64]:
```
df[df['Survived']==0]['Age'].max()
```

Out[64]:
```
74.0
```

In [65]:
```
df[df['Survived']==0]['Age'].min()
```

Out[65]:
```
1.0
```

In [66]:
```
df[df['Survived']==1]
```

Out[66]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
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	Heikkinen, 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
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
...	...	...	...	...	...	...	...	...	...	...	...	...
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15.0	0	0	2667	7.2250	NaN	C
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25.0	0	1	230433	26.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

342 rows × 12 columns

In [67]:
```
df[df['Survived']==1]['Age'].max()
```

Out[67]:
```
80.0
```

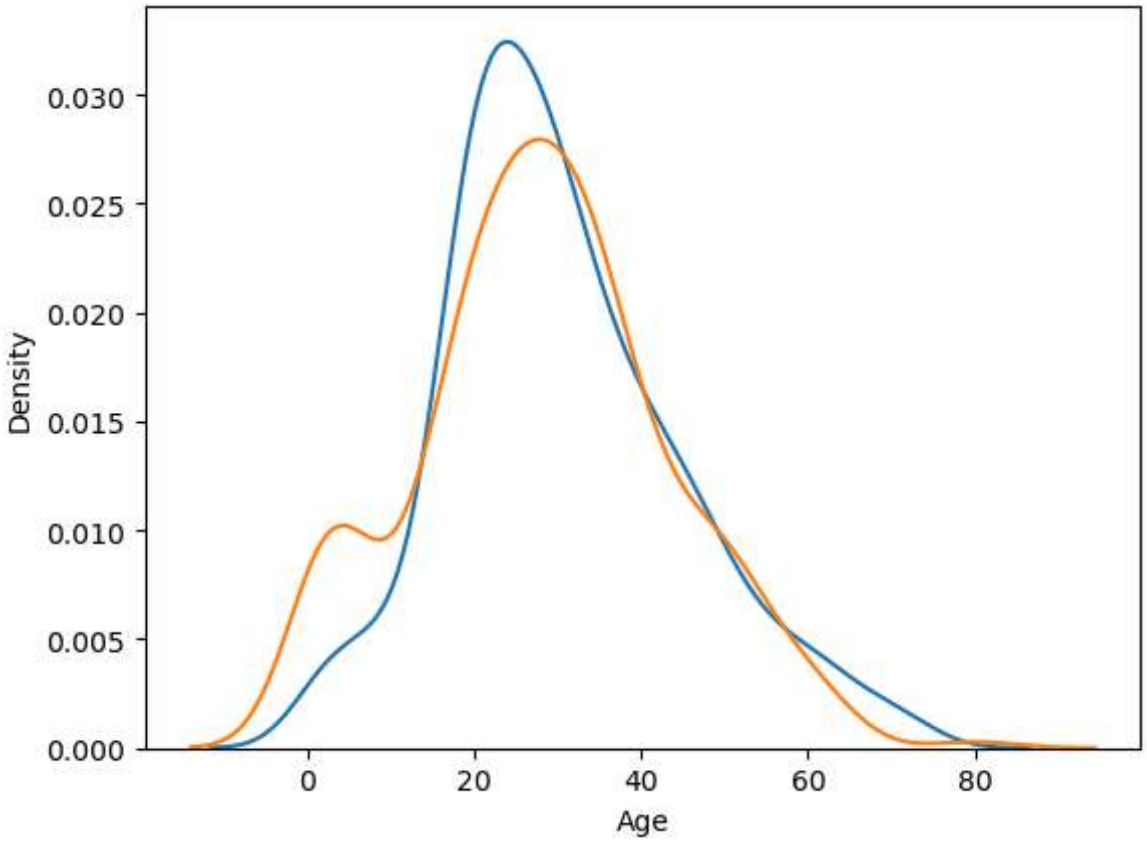
In [68]:
```
df[df['Survived']==1]['Age'].min()
```

Out[68]:
```
0.42
```

In [69]:
```
sns.distplot(df[df['Survived']==0]['Age'],hist=False)
sns.distplot(df[df['Survived']==1]['Age'],hist=False)
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

Out[69]: <AxesSubplot:xlabel='Age', ylabel='Density'>



Heat map on catagaerical varibles

In [70]:
```
df.head()
```

Out[70]:

	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	Heikkinen, 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

In [71]:
```
df['Pclass']
```

Out[71]:
```
0      3
1      1
2      3
3      1
4      3
      ..
886    2
887    1
888    3
889    1
890    3
Name: Pclass, Length: 891, dtype: int64
```

In [72]:
```
df['Survived']
```

Out[72]:
```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

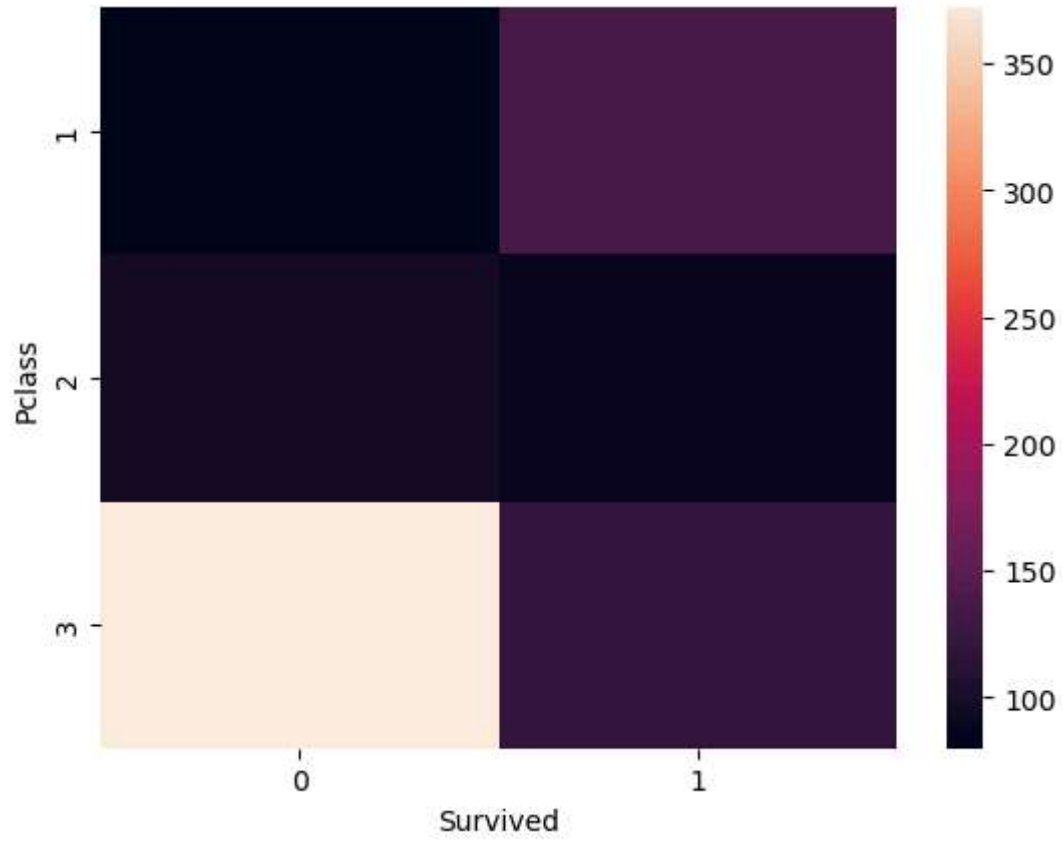
In [73]:
```
pd.crosstab(df['Pclass'],df['Survived'])
```

Out[73]:

Survived	0	1
Pclass		
1	80	136
2	97	87
3	372	119

In [74]:
```
sns.heatmap(pd.crosstab(df['Pclass'],df['Survived']))
```

Out[74]:
```
<AxesSubplot:xlabel='Survived', ylabel='Pclass'>
```



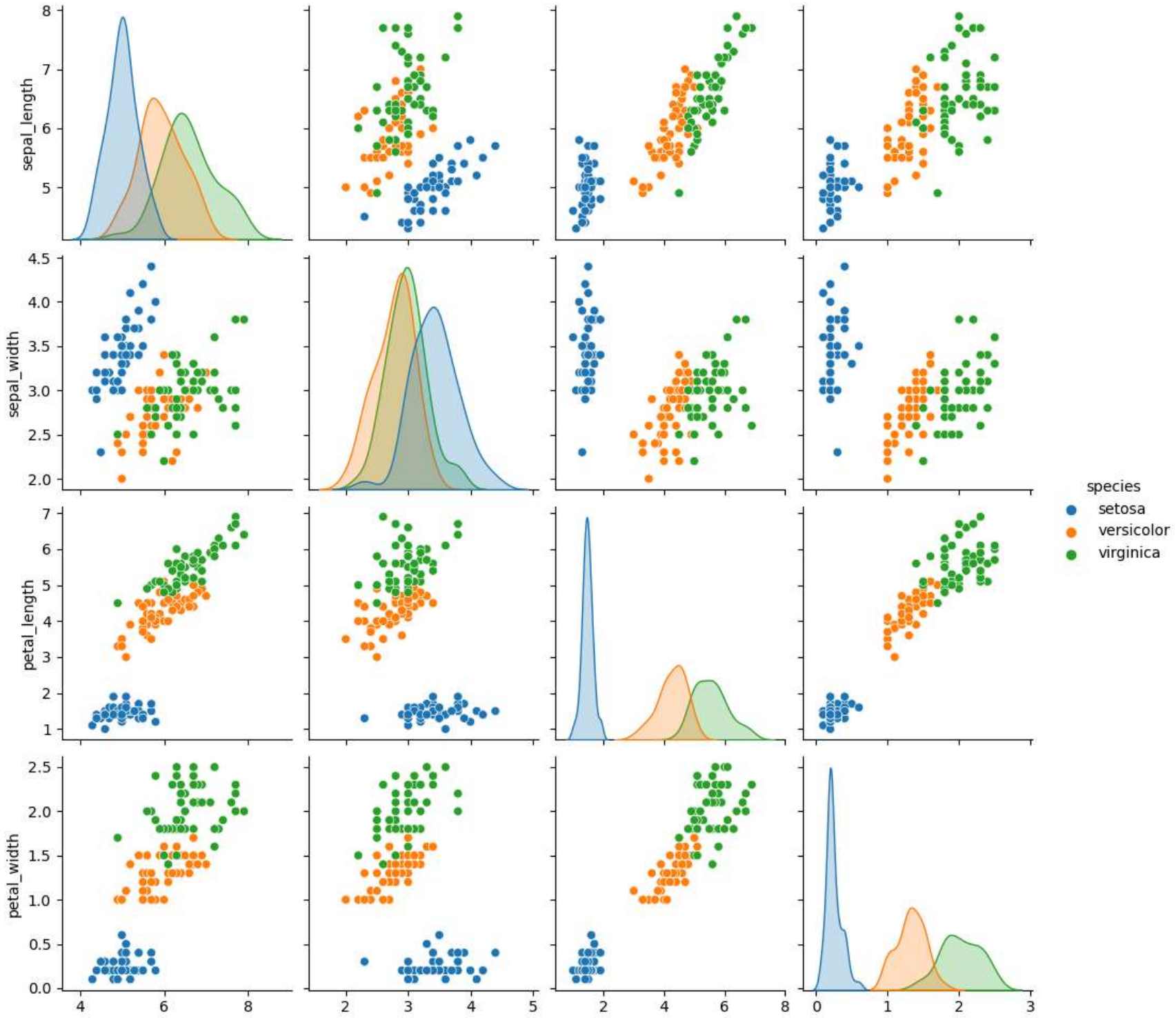
In [77]:
```
iris.head()
```

Out[77]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [79]:
```
sns.pairplot(iris,hue='species')
```

Out[79]:
```
<seaborn.axisgrid.PairGrid at 0x1ec869aeca0>
```

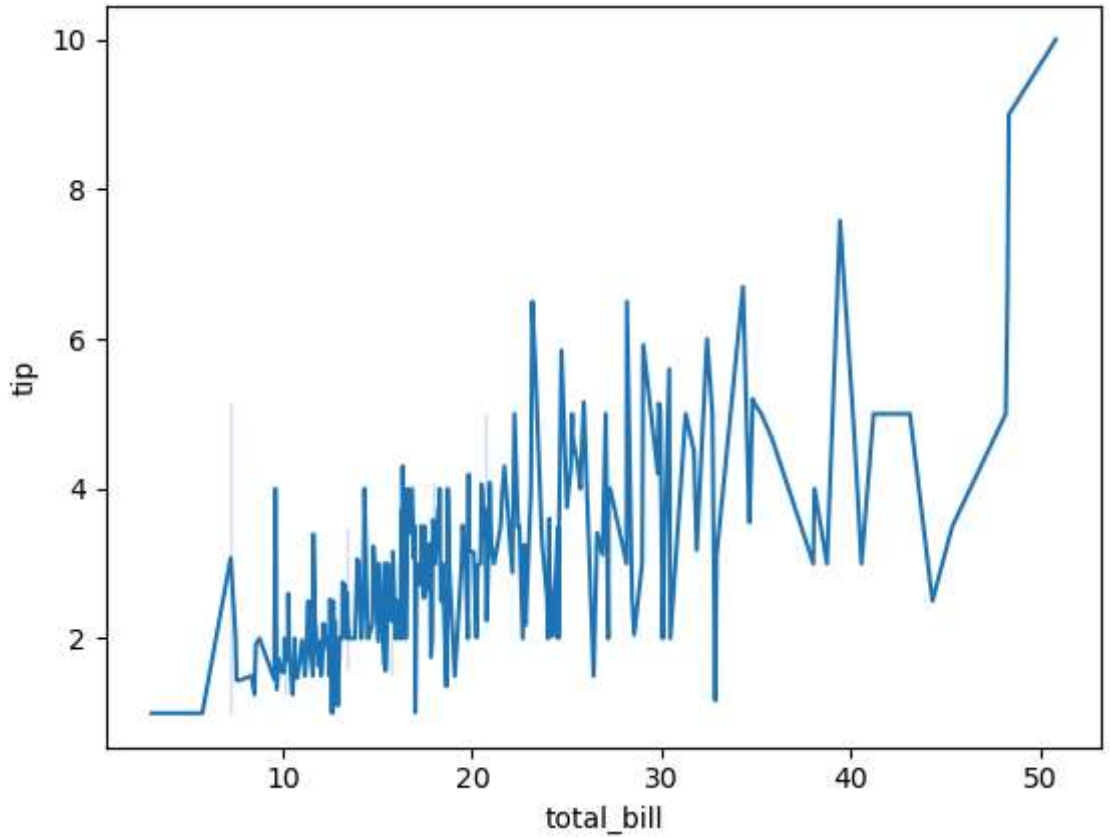

sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica

sepal_length sepal_width petal_length petal_width

In [80]:
```
sns.lineplot(tips['total_bill'],tips['tip'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[80]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



In [82]:
```
flights.head()
```

Out[82]:

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121

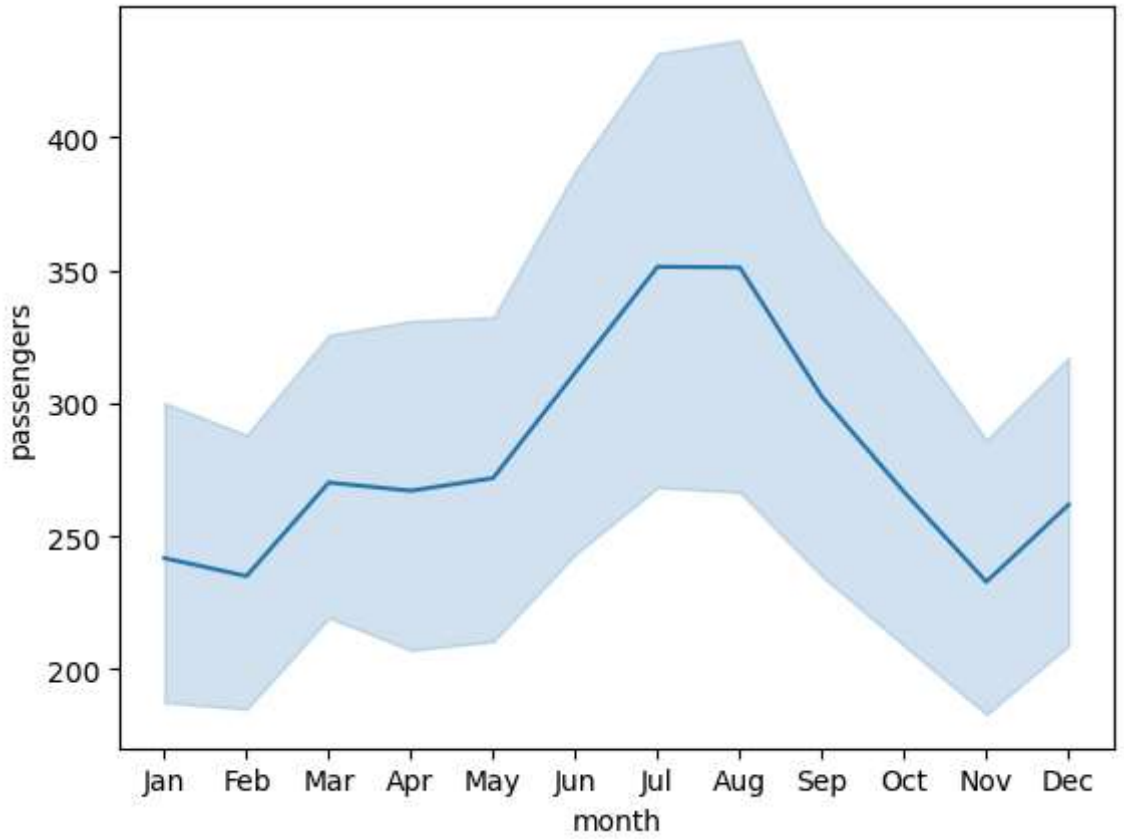
In [85]:
```
flights.groupby('year')
```

Out[85]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001EC87975100>
```

In [84]:
```
sns.lineplot(flights['month'],flights['passengers'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[84]:
```
<AxesSubplot:xlabel='month', ylabel='passengers'>
```



In [93]:
```
flight_new=flights.groupby('year').sum().reset_index()
```

In [94]:
```
flight_new
```

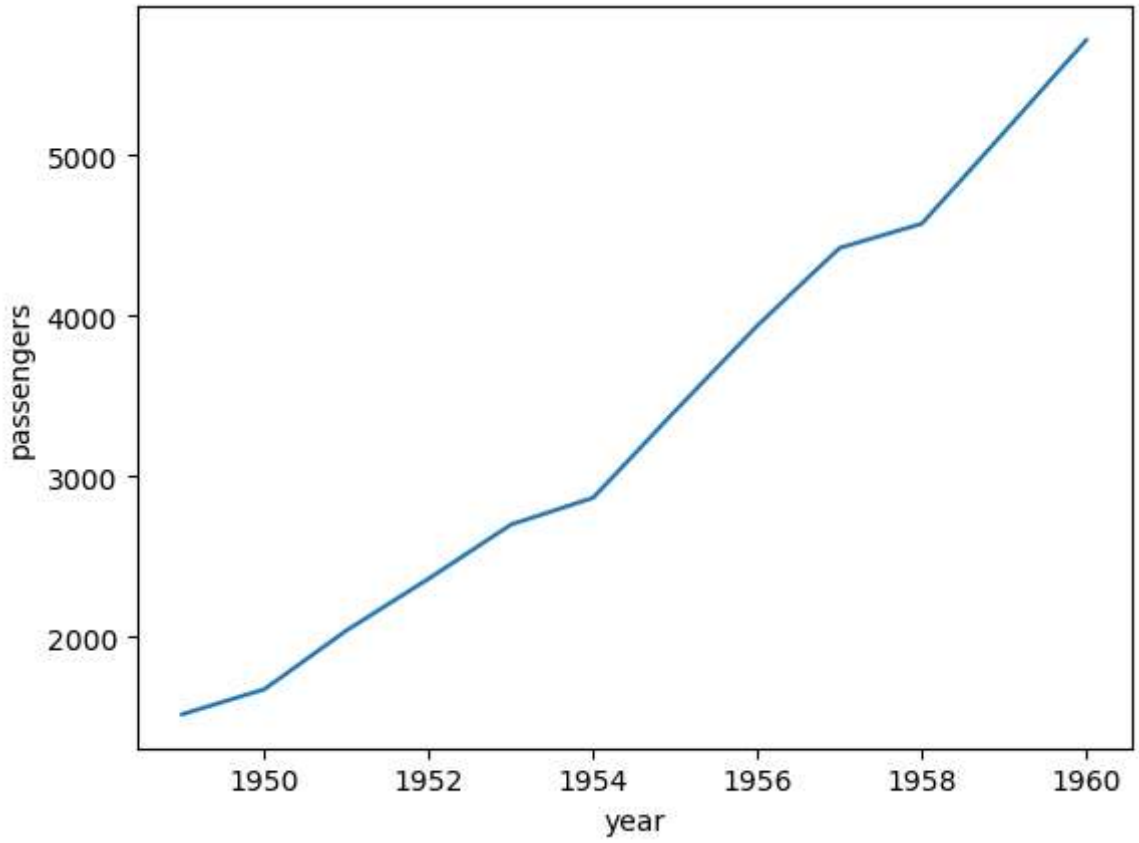
Out[94]:

	year	passengers
0	1949	1520
1	1950	1676
2	1951	2042
3	1952	2364
4	1953	2700
5	1954	2867
6	1955	3408
7	1956	3939
8	1957	4421
9	1958	4572
10	1959	5140
11	1960	5714

In [95]:
```
sns.lineplot(flight_new['year'],flight_new['passengers'])
```

```
C:\Users\BASAPARAJ\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[95]:
```
<AxesSubplot:xlabel='year', ylabel='passengers'>
```



Pivot Table in python

In [98]:
```
flights.head()
```

Out[98]:

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121

In [104…
```
flights.pivot_table(values='passengers',index='month',columns='year')
```

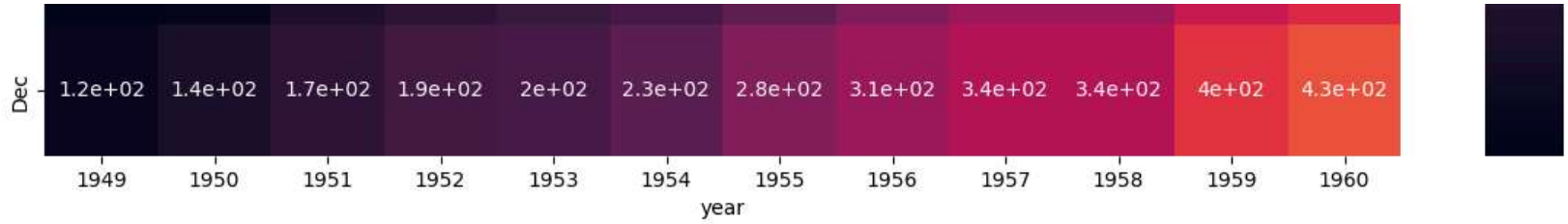
Out[104]:

year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
Jan	112	115	145	171	196	204	242	284	315	340	360	417
Feb	118	126	150	180	196	188	233	277	301	318	342	391
Mar	132	141	178	193	236	235	267	317	356	362	406	419
Apr	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
Jun	135	149	178	218	243	264	315	374	422	435	472	535
Jul	148	170	199	230	264	302	364	413	465	491	548	622
Aug	148	170	199	242	272	293	347	405	467	505	559	606
Sep	136	158	184	209	237	259	312	355	404	404	463	508
Oct	119	133	162	191	211	229	274	306	347	359	407	461
Nov	104	114	146	172	180	203	237	271	305	310	362	390
Dec	118	140	166	194	201	229	278	306	336	337	405	432

In [114…
```
plt.figure(figsize=(15,14))
sns.heatmap(flights.pivot_table(values='passengers',index='month',columns='year'),annot=True)
```

Out[114]:
```
<AxesSubplot:xlabel='year', ylabel='month'>
```


month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
1.1e+02 1.2e+02 1.4e+02 1.7e+02 2e+02 2e+02 2.4e+02 2.8e+02 3.2e+02 3.4e+02 3.6e+02 4.2e+02
1.2e+02 1.3e+02 1.5e+02 1.8e+02 2e+02 1.9e+02 2.3e+02 2.8e+02 3e+02 3.2e+02 3.4e+02 3.9e+02
1.3e+02 1.4e+02 1.8e+02 1.9e+02 2.4e+02 2.4e+02 2.7e+02 3.2e+02 3.6e+02 3.6e+02 4.1e+02 4.2e+02
1.3e+02 1.4e+02 1.6e+02 1.8e+02 2.4e+02 2.3e+02 2.7e+02 3.1e+02 3.5e+02 3.5e+02 4e+02 4.6e+02
1.2e+02 1.2e+02 1.7e+02 1.8e+02 2.3e+02 2.3e+02 2.7e+02 3.2e+02 3.6e+02 3.6e+02 4.2e+02 4.7e+02
1.4e+02 1.5e+02 1.8e+02 2.2e+02 2.4e+02 2.6e+02 3.2e+02 3.7e+02 4.2e+02 4.4e+02 4.7e+02 5.4e+02
1.5e+02 1.7e+02 2e+02 2.3e+02 2.6e+02 3e+02 3.6e+02 4.1e+02 4.6e+02 4.9e+02 5.5e+02 6.2e+02
1.5e+02 1.7e+02 2e+02 2.4e+02 2.7e+02 2.9e+02 3.5e+02 4e+02 4.7e+02 5e+02 5.6e+02 6.1e+02
1.4e+02 1.6e+02 1.8e+02 2.1e+02 2.4e+02 2.6e+02 3.1e+02 3.6e+02 4e+02 4e+02 4.6e+02 5.1e+02
1.2e+02 1.3e+02 1.6e+02 1.9e+02 2.1e+02 2.3e+02 2.7e+02 3.1e+02 3.5e+02 3.6e+02 4.1e+02 4.6e+02
1e+02 1.1e+02 1.5e+02 1.7e+02 1.8e+02 2e+02 2.4e+02 2.7e+02 3e+02 3.1e+02 3.6e+02 3.9e+02
600
500
400
300
200


Dec
1.2e+02
1.4e+02
1.7e+02
1.9e+02
2e+02
2.3e+02
2.8e+02
3.1e+02
3.4e+02
3.4e+02
4e+02
4.3e+02
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
year

In [ ]: