```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data=pd.read_csv('train.csv')
```

```
data
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
				Allen, Mr.						

Next steps: Generate code with data | View recommended plots | New interactive sheet

1. Display Top 5 Rows of The Dataset

```
data.head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833

Next steps: Generate code with data | View recommended plots | New interactive sheet

2. Check the Last 3 Rows of The Dataset

```
data.tail(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
888	889	0	3	Johnston, Miss. Catherine Helen	female	NaN	1	2	W./C. 6607	23.45	

3. Find Shape of Our Dataset (Number of Rows & Number of Columns)

```
data.shape
```

```
(891, 12)
```

```
print("Number of Row:",data.shape[0])
print("Number of Columns :",data.shape[1])
```

```
Number of Row: 891
Number of Columns : 12
```

4. Get Information About Our Dataset Like Total Number Rows, Total Number of Columns, Datatypes of Each Column And Memory Requirement

```
data.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
```

5. Get Overall Statistics About The Dataframe

```
data.describe(include='all')
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000
unique	NaN	NaN	NaN	891	2	NaN	NaN	N
top	NaN	NaN	NaN	Dooley, Mr. Patrick	male	NaN	NaN	N
freq	NaN	NaN	NaN	1	577	NaN	NaN	N
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000

6. Data Filtering

```
data.columns
```

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

```
data[['Name','Age']]
```

	Name	Age
0	Braund, Mr. Owen Harris	22.0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0
2	Heikkinen, Miss. Laina	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0
4	Allen, Mr. William Henry	35.0
...	...	...
886	Montvila, Rev. Juozas	27.0
887	Graham, Miss. Margaret Edith	19.0
888	Johnston, Miss. Catherine Helen "Carrie"	NaN
889	Behr, Mr. Karl Howell	26.0
890	Dooley, Mr. Patrick	32.0

891 rows × 2 columns

```
sum(data['Sex']=='male')
```

```
577
```

```
data[data['Sex']=='male'].head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	N

```
data.columns
```

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

```
data['Survived'].value_counts()
```

Survived	count
0	549
1	342

dtype: int64

```
data[data['Survived']==1]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.13

7. Check Null Values In The Dataset

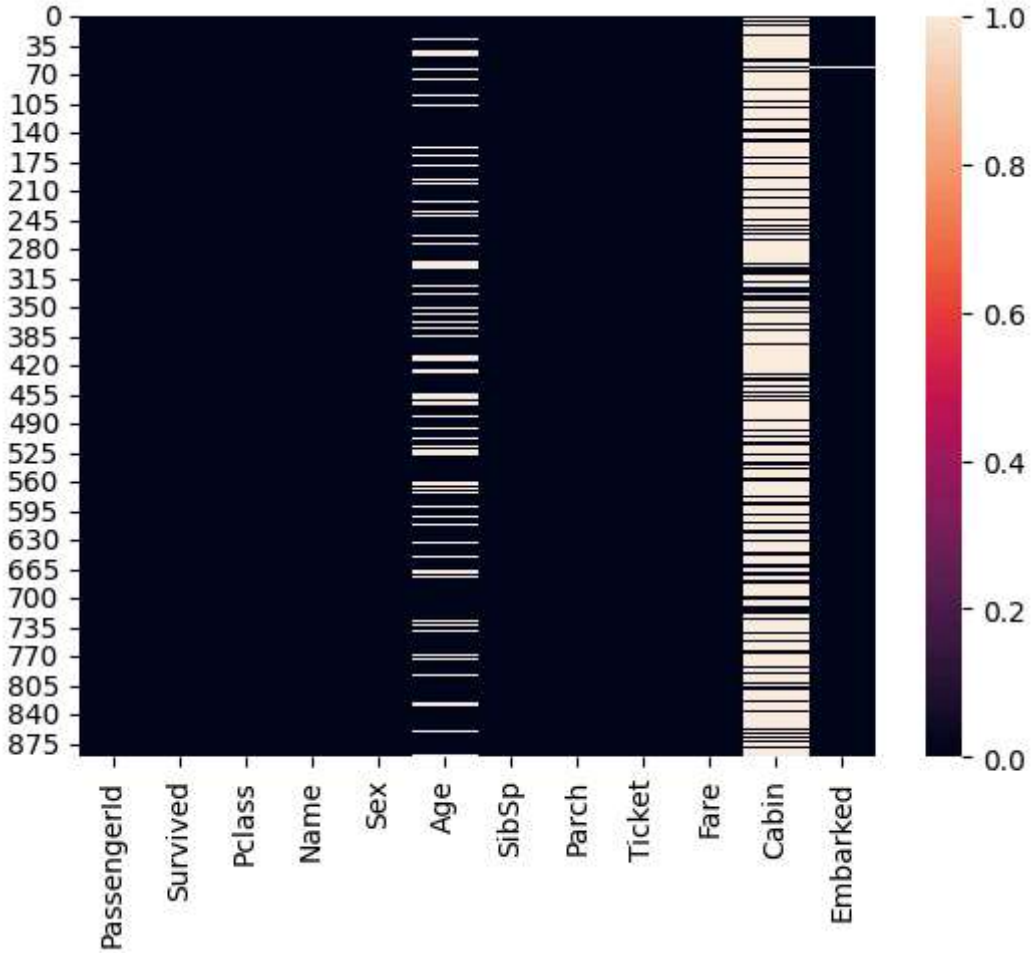
```
data.isnull().sum()
```

	0
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

```
sns.heatmap(data.isnull())
```

```
<Axes: >
```




```
per_missing = data.isnull().sum() * 100 / len(data)
per_missing
```

	0
PassengerId	0.000000
Survived	0.000000
Pclass	0.000000
Name	0.000000
Sex	0.000000
Age	19.865320
SibSp	0.000000
Parch	0.000000
Ticket	0.000000
Fare	0.000000
Cabin	77.104377
Embarked	0.224467

dtype: float64

8. Drop the Column

```
data.drop('Cabin',axis=1,inplace=True)
```

```
data.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Embarked	2

dtype: int64

9. Handle Missing Values

```
data.columns
```

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

```
#We will handle null Values in Embarked Column
data['Embarked']
```

	Embarked
0	S
1	C
2	S
3	S
4	S
...	...
886	S
887	S
888	S
889	C
890	Q

891 rows × 1 columns

dtype: object

```
data['Embarked'].mode()
```

	Embarked
0	S

dtype: object

```
data['Embarked'].fillna('S',inplace=True)
```

```
data.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Embarked	0

dtype: int64

```
# we will handle null values in Age Column
data['Age']
```

	Age
0	22.0
1	38.0
2	26.0
3	35.0
4	35.0
...	...
886	27.0
887	19.0
888	NaN
889	26.0
890	32.0

891 rows × 1 columns

dtype: float64

```
data['Age'].fillna(data['Age'].mean(),inplace=True)
```

```
data.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Embarked	0

dtype: int64

10. Categorical Data Encoding

```
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	71.2833

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data['Sex'].unique()
```

```
array(['male', 'female'], dtype=object)
```

```
data['Gender']=data['Sex'].map({'male':1,'female':0})
```

```
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
x=data['Sex'].map({'male':1,'female':0})
```

```
data.insert(5,'Gender_New',x)
```

```
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Gender_New	Age	SibSp	Parch	Ti
0	1	0	3	Braund, Mr. Owen Harris	male	1	22.0	1	0	A/5 2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	0	38.0	1	0	PC 17

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data['Embarked'].unique()
```

```
array(['S', 'C', 'Q'], dtype=object)
```

```
pd.get_dummies(data,columns=['Embarked'],drop_first=True)
```

	PassengerId	Survived	Pclass	Name	Sex	Gender_New	Age	SibSp	Parch
0	1	0	3	Braund, Mr. Owen Harris	male	1	22.000000	1	0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	0	38.000000	1	0
2	3	1	3	Heikkinen, Miss. Laina	female	0	26.000000	0	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	0	35.000000	1	0
4	5	0	3	Allen, Mr. William Henry	male	1	35.000000	0	0
...	...	...	...	...	...	...	...	...	..
886	887	0	2	Montvila, Rev. Juozas	male	1	27.000000	0	0
887	888	1	1	Graham, Miss. Margaret Edith	female	0	19.000000	0	0

```
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Gender_New	Age	SibSp	Parch	Tic
0	1	0	3	Braund, Mr. Owen Harris	male	1	22.0	1	0	A/5 2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	0	38.0	1	0	PC 17

Next steps: Generate code with data | View recommended plots | New interactive sheet

11. What is Univariate Analysis?

- How Many People Survived And How Many Died?
- How Many Passengers Were In First Class, Second Class, and Third Class?
- Number of Male And Female Passengers

```
data.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Gender_New', 'Age',
       'SibSp', 'Parch', 'Ticket', 'Fare', 'Embarked', 'Gender'],
      dtype='object')
```

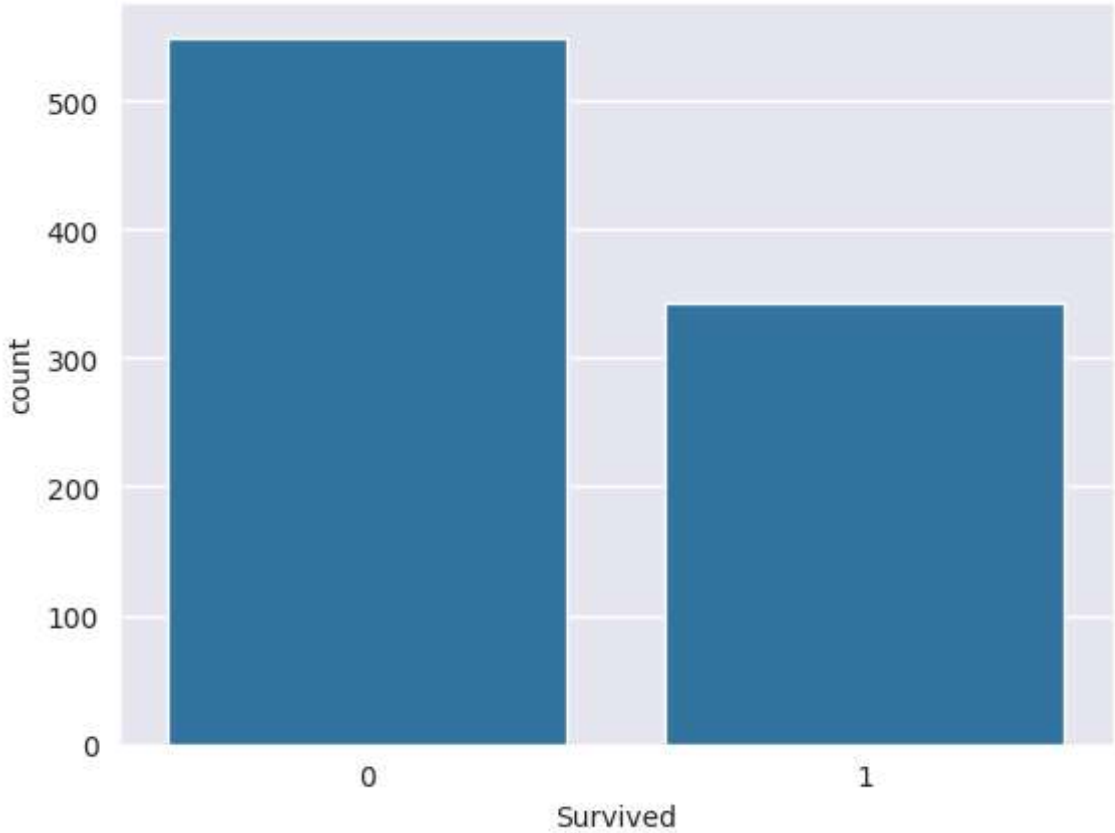
```
#How Many People Survived And How Many Died?
data['Survived'].value_counts()
```

Survived	count
0	549
1	342

dtype: int64

```
sns.countplot(x="Survived", data=data)
```

```
<Axes: xlabel='Survived', ylabel='count'>
```



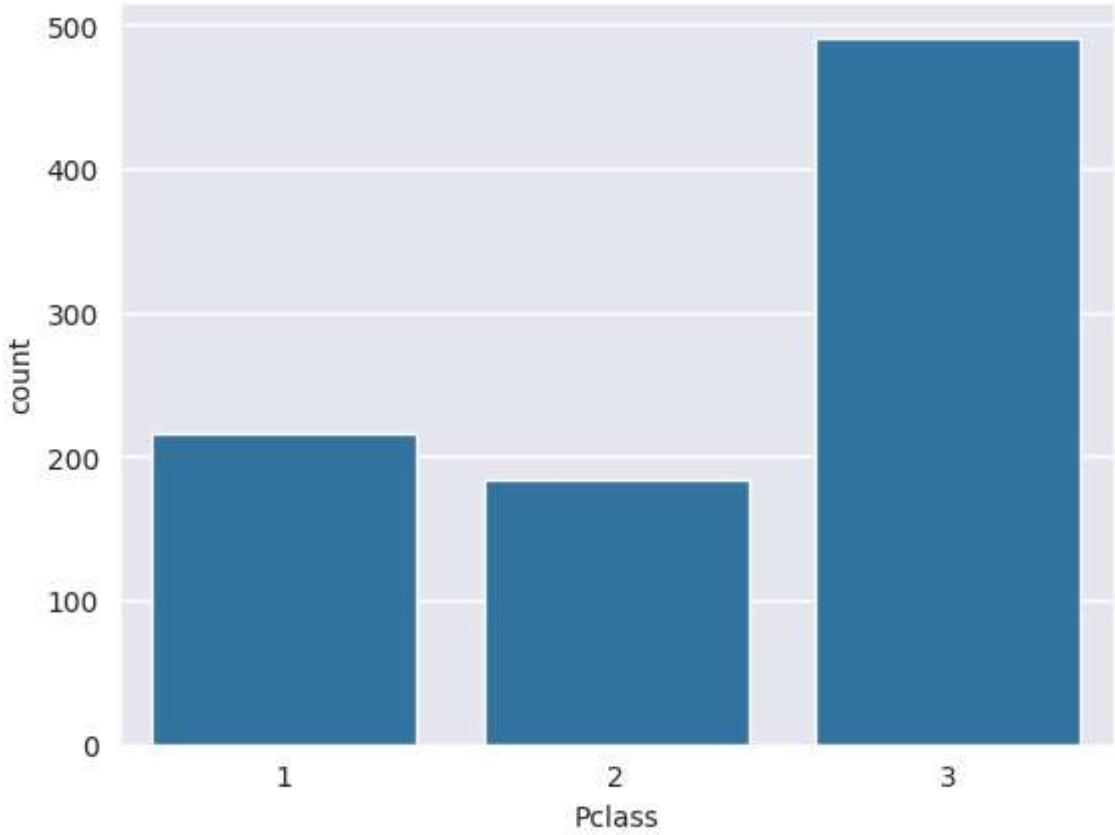

```
#How Many Passengers Were In First Class, Second Class, and Third Class?
data['Pclass'].value_counts()
```

Pclass	count
3	491
1	216
2	184

dtype: int64

```
sns.countplot(x="Pclass", data=data)
```

```
<Axes: xlabel='Pclass', ylabel='count'>
```



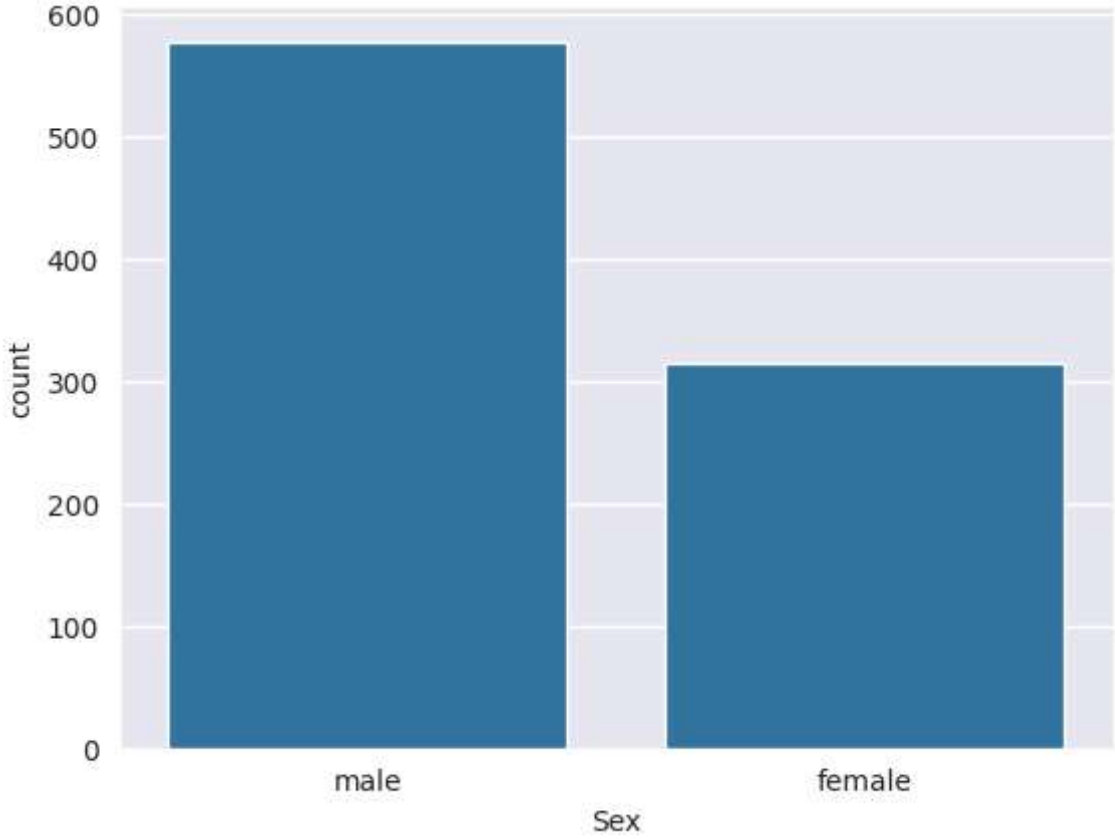

```
# Number of Male And Female Passengers
data['Sex'].value_counts()
```

Sex	count
male	577
female	314

dtype: int64

```
sns.countplot(x="Sex", data=data)
```

```
<Axes: xlabel='Sex', ylabel='count'>
```




```
plt.hist(data['Age'])
```

```
(array([ 54.,  46., 177., 346., 118.,  70.,  45.,  24.,   9.,   2.]),
 array([ 0.42,   8.378,  16.336,  24.294,  32.252,  40.21,  48.168,  56.126
```