```
import numpy as np
import pandas as pd
import seaborn as sns
```

```
df = sns.load_dataset('titanic')
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

```
df.columns
```

```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

```
df.describe()
```

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

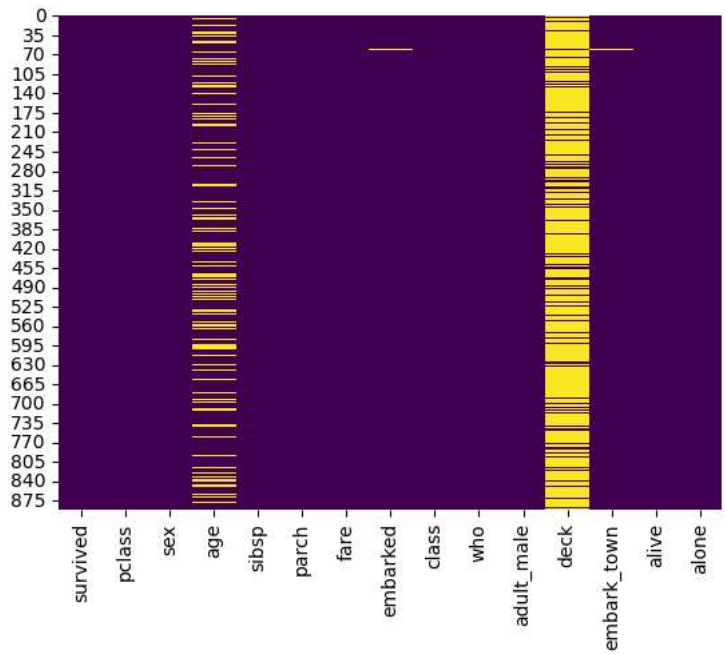
```
df.isnull().sum()
```

```
survived         0
pclass           0
sex              0
```

```
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

```
sns.heatmap(df.isnull(),cbar = False, cmap = 'viridis')
```

```
<Axes: >
```

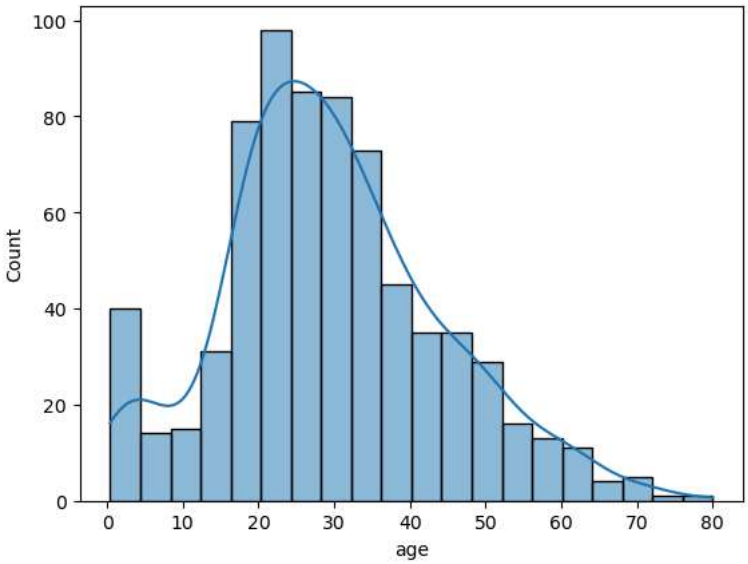



```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

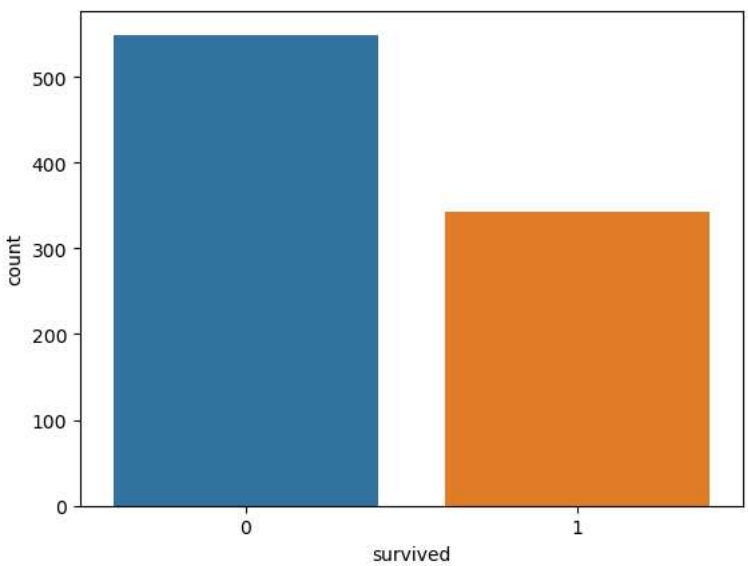
```
sns.histplot(df['age'],kde = True)
```

```
C:\Users\wwwsi\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be remov
  with pd.option_context('mode.use_inf_as_na', True):
<Axes: xlabel='age', ylabel='Count'>
```




```
sns.countplot(data = df, x = 'survived')
```

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



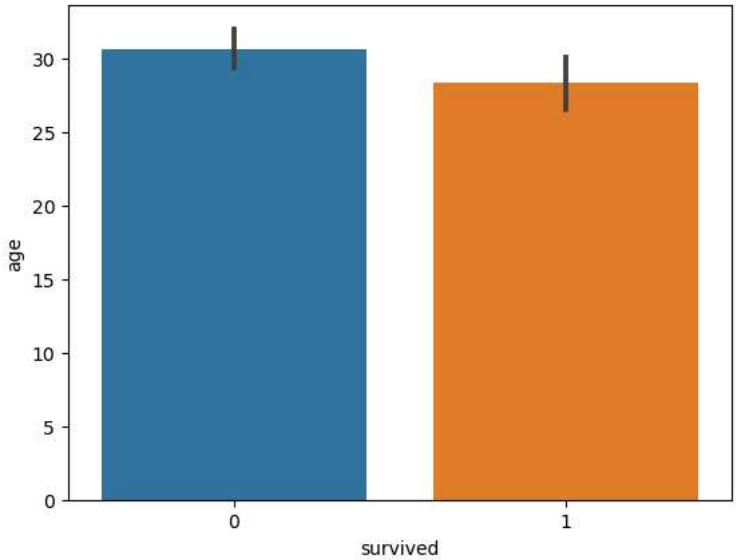

```
import plotly.express as px
```

```
new_df = df['survived'].value_counts().reset_index()
```

Start coding or generate with AI.

```
sns.barplot(data = df, x = 'survived', y = 'age')
```

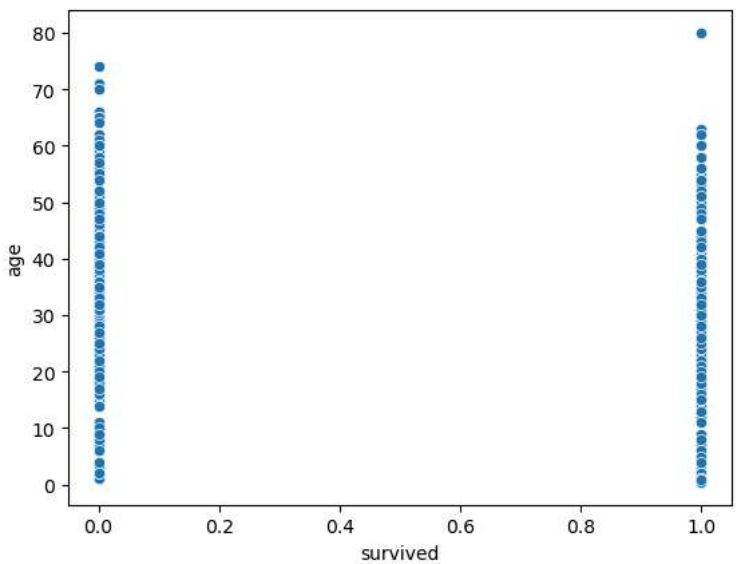
```
<Axes: xlabel='survived', ylabel='age'>
```



Start coding or generate with AI.

```
sns.scatterplot(data = df, x = 'survived', y = 'age')
```

```
<Axes: xlabel='survived', ylabel='age'>
```



Double-click (or enter) to edit

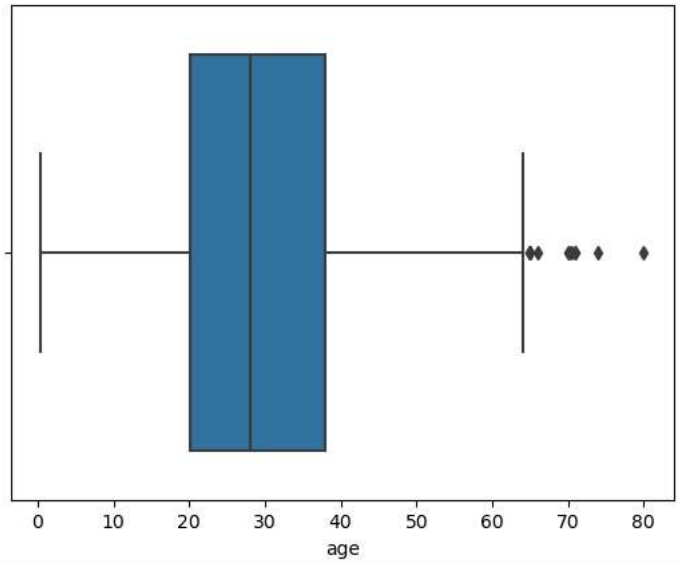
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
```

```
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

```
sns.boxplot(data = df, x = 'age')
```

```
<Axes: xlabel='age'>
```




```
df.columns
```

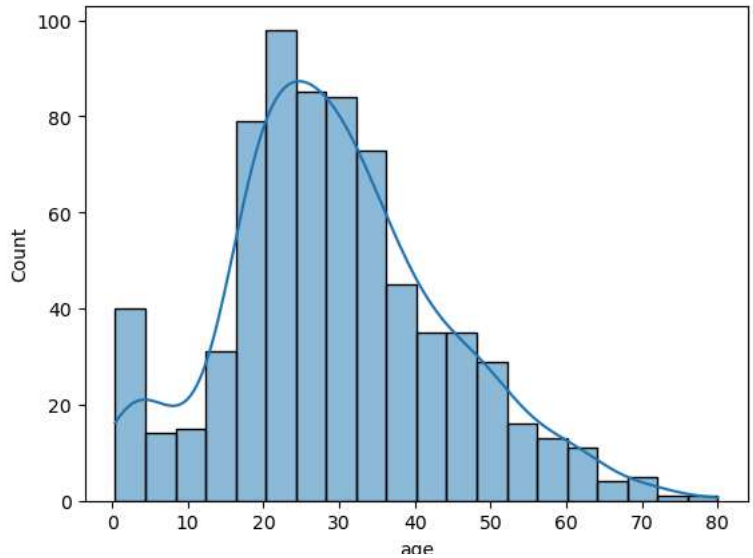
```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

```
# Data preprocessing

from sklearn.preprocessing import LabelEncoder


# Handling the missing values

df.isnull().sum()
```

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

```
sns.histplot(df['age'],kde = True)
```

```
C:\Users\wwwsi\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be remov
  with pd.option_context('mode.use_inf_as_na', True):
<Axes: xlabel='age', ylabel='Count'>
```




```
df['age'].fillna(df['age']).median()
```

```
28.0
```

```
df['age'].isnull().sum()
```

```
177
```

```
df['age'] = df['age'].fillna(df['age']).median()
```

```
df.isnull().sum()
```

```
survived         0
pclass           0
sex              0
age              0
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

Double-click (or enter) to edit

```
# encoding
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	28.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	28.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	28.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	28.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	28.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

```
# label encoding

df['sex'] = df['sex'].map({"male" : 0, "female": 1})
```

```
df['sex'].head(5)
```

```
0    0
1    1
2    1
3    1
4    0
Name: sex, dtype: int64
```

```
# outliers

df['age'].head(5)
```

```
0    28.0
1    28.0
2    28.0
3    28.0
4    28.0
Name: age, dtype: float64
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	0	28.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	1	28.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	1	28.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	1	28.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	0	28.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

```
df.drop('class', axis = 1, inplace = True)
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	who	adult_male	deck	embark_town	alive	alone
0	0	3	0	28.0	1	0	7.2500	S	man	True	NaN	Southampton	no	False
1	1	1	1	28.0	1	0	71.2833	C	woman	False	C	Cherbourg	yes	False
2	1	3	1	28.0	0	0	7.9250	S	woman	False	NaN	Southampton	yes	True
3	1	1	1	28.0	1	0	53.1000	S	woman	False	C	Southampton	yes	False
4	0	3	0	28.0	0	0	8.0500	S	man	True	NaN	Southampton	no	True

```
df.drop('embark_town',axis = 1, inplace = True)
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	who	adult_male	deck	alive	alone
0	0	3	0	28.0	1	0	7.2500	S	man	True	NaN	no	False
1	1	1	1	28.0	1	0	71.2833	C	woman	False	C	yes	False
2	1	3	1	28.0	0	0	7.9250	S	woman	False	NaN	yes	True
3	1	1	1	28.0	1	0	53.1000	S	woman	False	C	yes	False
4	0	3	0	28.0	0	0	8.0500	S	man	True	NaN	no	True

```
df['embarked'].unique()
```

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

```
df['who'].unique()
```

```
array(['man', 'woman', 'child'], dtype=object)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   survived    891 non-null    int64
 1   pclass      891 non-null    int64
 2   sex         891 non-null    int64
 3   age         891 non-null    float64
 4   sibsp       891 non-null    int64
 5   parch       891 non-null    int64
 6   fare        891 non-null    float64
 7   embarked    889 non-null    object
 8   who         891 non-null    object
 9   adult_male  891 non-null    bool
 10  deck        203 non-null    category
 11  alive       891 non-null    object
 12  alone       891 non-null    bool
dtypes: bool(2), category(1), float64(2), int64(5), object(3)
memory usage: 72.7+ KB
```

```
df.drop('deck', axis = 1, inplace = True)
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	who	adult_male	alive	alone
0	0	3	0	28.0	1	0	7.2500	S	man	True	no	False
1	1	1	1	28.0	1	0	71.2833	C	woman	False	yes	False
2	1	3	1	28.0	0	0	7.9250	S	woman	False	yes	True
3	1	1	1	28.0	1	0	53.1000	S	woman	False	yes	False
4	0	3	0	28.0	0	0	8.0500	S	man	True	no	True

```
df.drop('adult_male',axis = 1, inplace = True)
```

```
df.drop('alive',axis = 1, inplace = True)
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	who	alone
0	0	3	0	28.0	1	0	7.2500	S	man	False
1	1	1	1	28.0	1	0	71.2833	C	woman	False
2	1	3	1	28.0	0	0	7.9250	S	woman	True
3	1	1	1	28.0	1	0	53.1000	S	woman	False
4	0	3	0	28.0	0	0	8.0500	S	man	True

```
df.drop('alone',axis = 1, inplace = True)
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	who
0	0	3	0	28.0	1	0	7.2500	S	man
1	1	1	1	28.0	1	0	71.2833	C	woman
2	1	3	1	28.0	0	0	7.9250	S	woman
3	1	1	1	28.0	1	0	53.1000	S	woman
4	0	3	0	28.0	0	0	8.0500	S	man

```
df.isnull().sum()
```

```
survived    0
pclass      0
sex         0
age         0
sibsp       0
parch       0
fare        0
embarked    2
who         0
dtype: int64
```

```
df['embarked'].dropna(inplace = True)
```

```
df.dropna()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	who
0	0	3	0	28.0	1	0	7.2500	S	man
1	1	1	1	28.0	1	0	71.2833	C	woman
2	1	3	1	28.0	0	0	7.9250	S	woman
3	1	1	1	28.0	1	0	53.1000	S	woman
4	0	3	0	28.0	0	0	8.0500	S	man
...	...	...	...	...	...	...	...	...	...
886	0	2	0	28.0	0	0	13.0000	S	man
887	1	1	1	28.0	0	0	30.0000	S	woman
888	0	3	1	28.0	1	2	23.4500	S	woman
889	1	1	0	28.0	0	0	30.0000	C	man
890	0	3	0	28.0	0	0	7.7500	Q	man

889 rows × 9 columns

```
df['embarked'] = df['embarked'].fillna(df['embarked'].mode()[0])
```

```
df.isnull().sum()
```

```
survived    0
pclass      0
sex         0
age         0
sibsp       0
parch       0
fare        0
embarked    0
who         0
dtype: int64
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	who
0	0	3	0	28.0	1	0	7.2500	S	man
1	1	1	1	28.0	1	0	71.2833	C	woman
2	1	3	1	28.0	0	0	7.9250	S	woman
3	1	1	1	28.0	1	0	53.1000	S	woman
4	0	3	0	28.0	0	0	8.0500	S	man

```
df['embarked'].unique()
```

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

```
df['embarked'] = df['embarked'].map({'S':0, 'C' : 1, 'Q' : 2})
```

```
df['embarked'].unique()
```

```
array([0, 1, 2], dtype=int64)
```

```
df.head(5)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	who
0	0	3	0	28.0	1	0	7.2500	0	man
1	1	1	1	28.0	1	0	71.2833	1	woman
2	1	3	1	28.0	0	0	7.9250	0	woman
3	1	1	1	28.0	1	0	53.1000	0	woman
4	0	3	0	28.0	0	0	8.0500	0	man

Start coding or generate with AI.

```
df['who'] = df['who'].map({'man' : 0, 'woman': 1, 'child' : 2})
```

```
df.sample(5)
```

	pclass	sex	age	sibsp	parch	fare	embarked	who
246	3	1	28.0	0	0	7.7750	0	1
13	3	0	28.0	1	5	31.2750	0	0
144	2	0	28.0	0	0	11.5000	0	0
514	3	0	28.0	0	0	7.4958	0	0
271	3	0	28.0	0	0	0.0000	0	0

```
x=df.drop("survived",axis=1,inplace=True)
```

```
from sklearn.preprocessing import Normalizer
```