


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```


```
dat = pd.read_csv("/content/Titanic-Dataset.csv")
dat
```



	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
...	...	...	...		...	...	...	...	...		...	...	...	...
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

```
dat.head()
```



	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

```
dat.head(2)
```



	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

dat.shape

(891, 12)

dat.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
```

dat.isnull()

	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

dat.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

```
dat.isnull().sum() / dat.shape[0] * 100
```




	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

```
# dropping cabin column as its null value is more than 30%
```

```
dat.drop("Cabin", axis=1, inplace=True)
```

dat.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



```
#impute median value
```

```
dat.Age = dat.Age.fillna(dat.Age.median())
```

```
dat.Age.isnull().sum()
```



0

```
# Fare column, we drop the row
```

```
dat.dropna( inplace = True)
```

```
dat.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

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```


```
dat["Name"] = le.fit_transform(dat["Name"])
dat["Sex"] = le.fit_transform(dat["Sex"])
dat["Ticket"] = le.fit_transform(dat["Ticket"])
dat["Embarked"] = le.fit_transform(dat["Embarked"])
```

```
dat.info()
```



```
<class 'pandas.core.frame.DataFrame'>
Index: 889 entries, 0 to 890
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   PassengerId    889 non-null    int64  
 1   Survived       889 non-null    int64  
 2   Pclass        889 non-null    int64  
 3   Name          889 non-null    int64  
 4   Sex           889 non-null    int64  
 5   Age           889 non-null    float64 
 6   SibSp         889 non-null    int64  
 7   Parch         889 non-null    int64  
 8   Ticket        889 non-null    int64  
 9   Fare          889 non-null    float64 
10   Embarked      889 non-null    int64  
dtypes: float64(2), int64(9)
memory usage: 83.3 KB
```

```
dat.select_dtypes("int")
```




	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Embarked
0	1	0	3	108	1	1	0	522	2
1	2	1	1	190	0	1	0	595	0
2	3	1	3	353	0	0	0	668	2
3	4	1	1	272	0	1	0	48	2
4	5	0	3	15	1	0	0	471	2
...	...	...	...	...	...	...	...	...	...
886	887	0	2	547	1	0	0	100	2
887	888	1	1	303	0	0	0	14	2
888	889	0	3	412	0	1	2	674	2
889	890	1	1	81	1	0	0	8	0
890	891	0	3	220	1	0	0	465	1

889 rows × 9 columns

```
from sklearn.preprocessing import OneHotEncoder
```


```
ohe = OneHotEncoder()
```

```
ohe.fit_transform(dat[["Sex"]])
```



```
<889x2 sparse matrix of type '<class 'numpy.float64'>'
  with 889 stored elements in Compressed Sparse Row format>
```


```
ohe.fit_transform(dat[["Sex"]]).toarray()
```

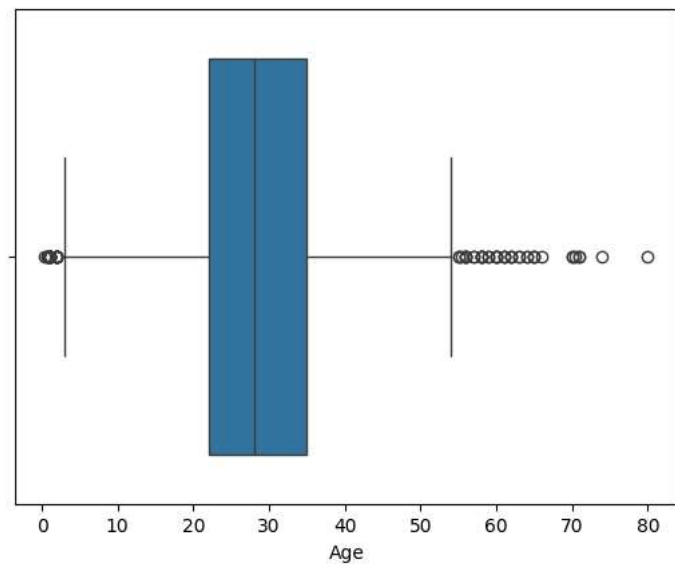


```
array([[0., 1.],
       [1., 0.],
       [1., 0.],
       ...,
       [1., 0.],
       [0., 1.],
       [0., 1.]])
```


```
import seaborn as sns
```

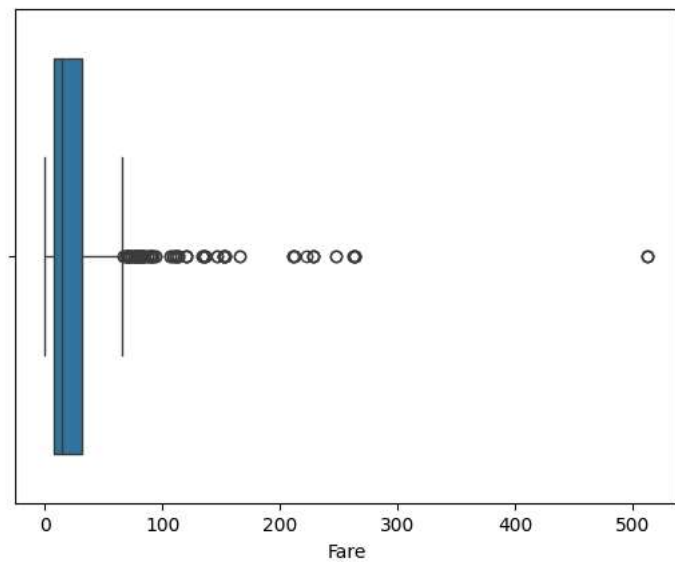
```
sns.boxplot(x=dat.Age)
```

 <Axes: xlabel='Age'>




```
sns.boxplot(x=dat.Fare)
```

 <Axes: xlabel='Fare'>



```
sns.distplot(dat.Fare)
```

 <ipython-input-31-273d31d0fb11>:1: UserWarning:

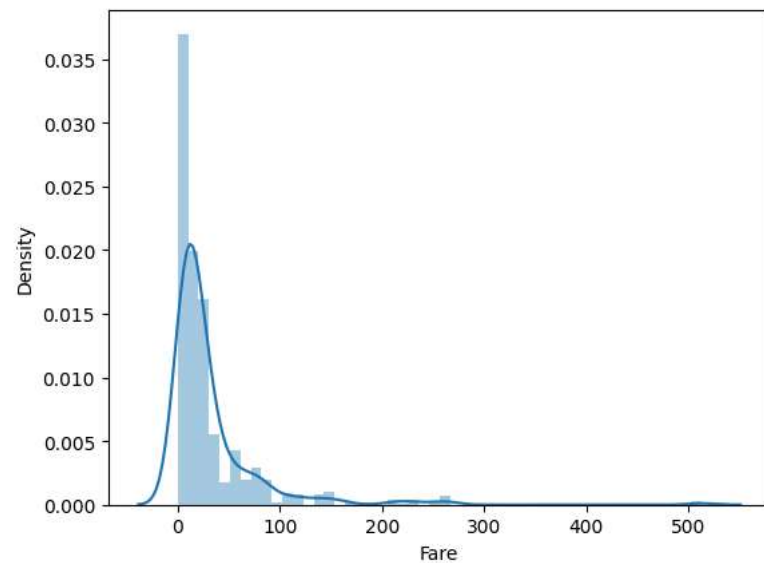
``distplot`` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either ``displot`` (a figure-level function with similar flexibility) or ``histplot`` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(dat.Fare)
<Axes: xlabel='Fare', ylabel='Density'>
```



```
sns.distplot(dat.Age)
```



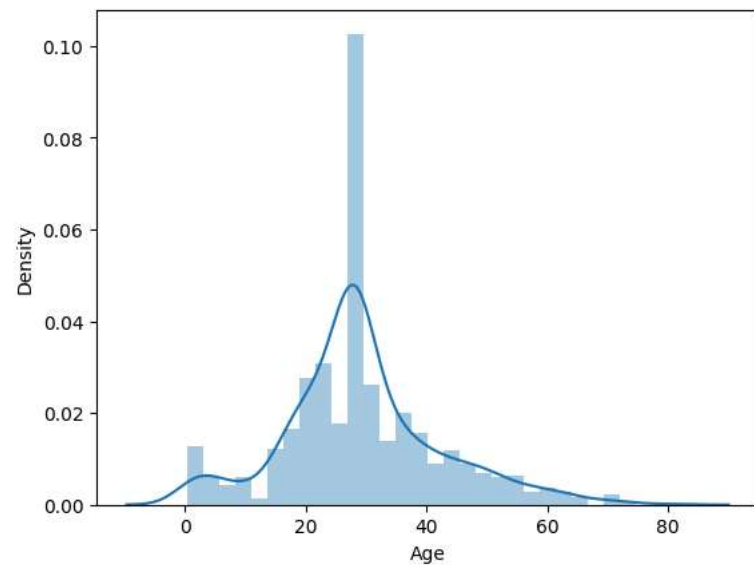
 <ipython-input-32-9ac54441ec7c>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

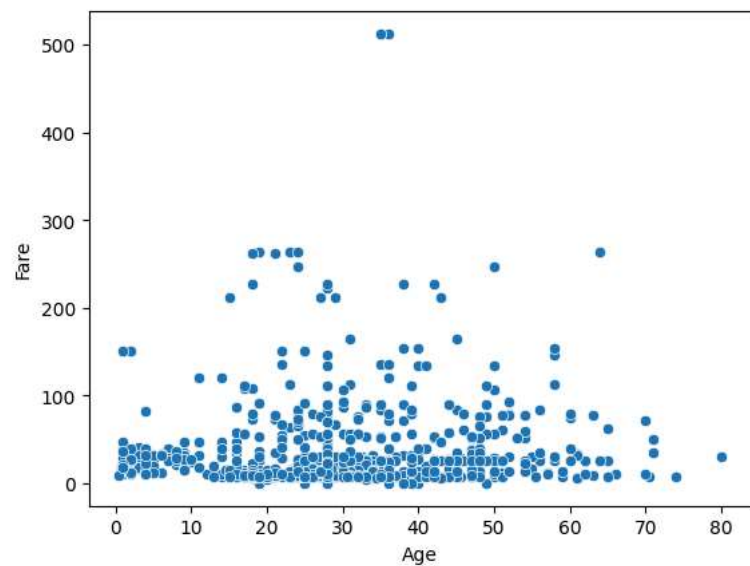
For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(dat.Age)
<Axes: xlabel='Age', ylabel='Density'>
```



```
sns.scatterplot(x=dat.Age, y=dat.Fare)
```

<Axes: xlabel='Age', ylabel='Fare'>



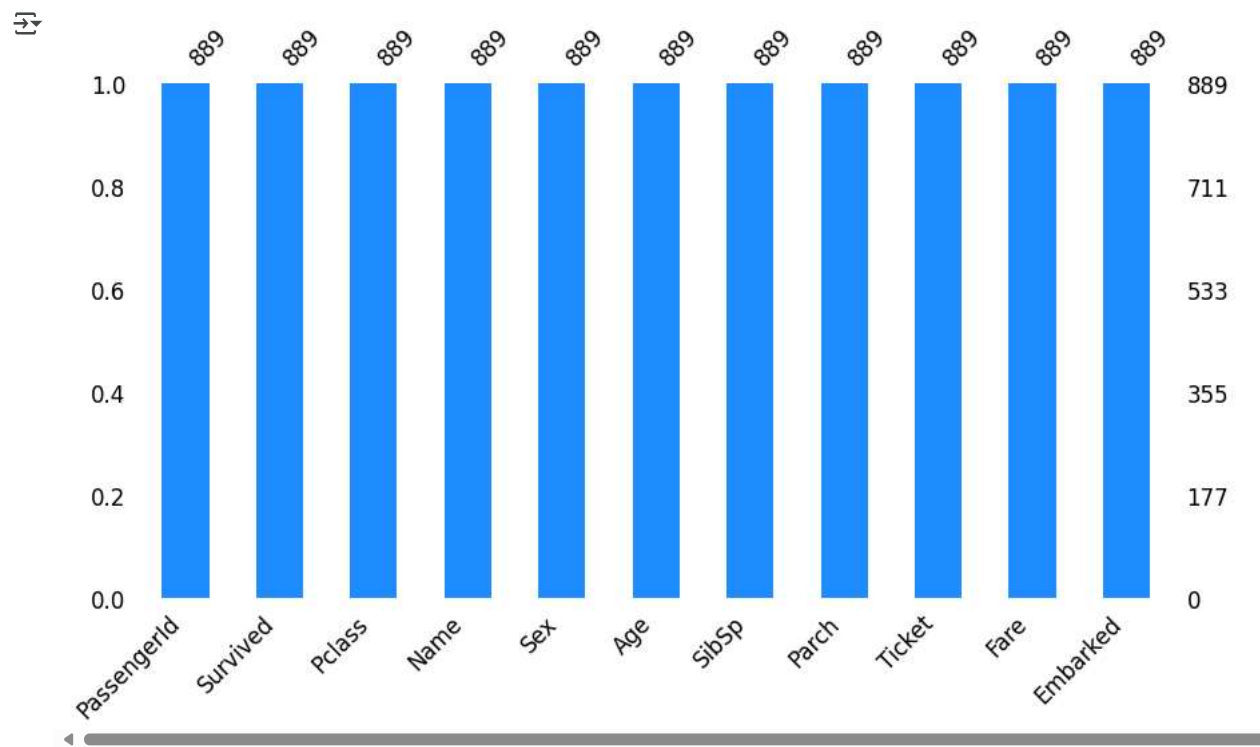
```
dat['Embarked'] = dat['Embarked'].fillna(method='bfill')
```

```
import missingno as msno
msno.matrix(dat)
plt.show()
```

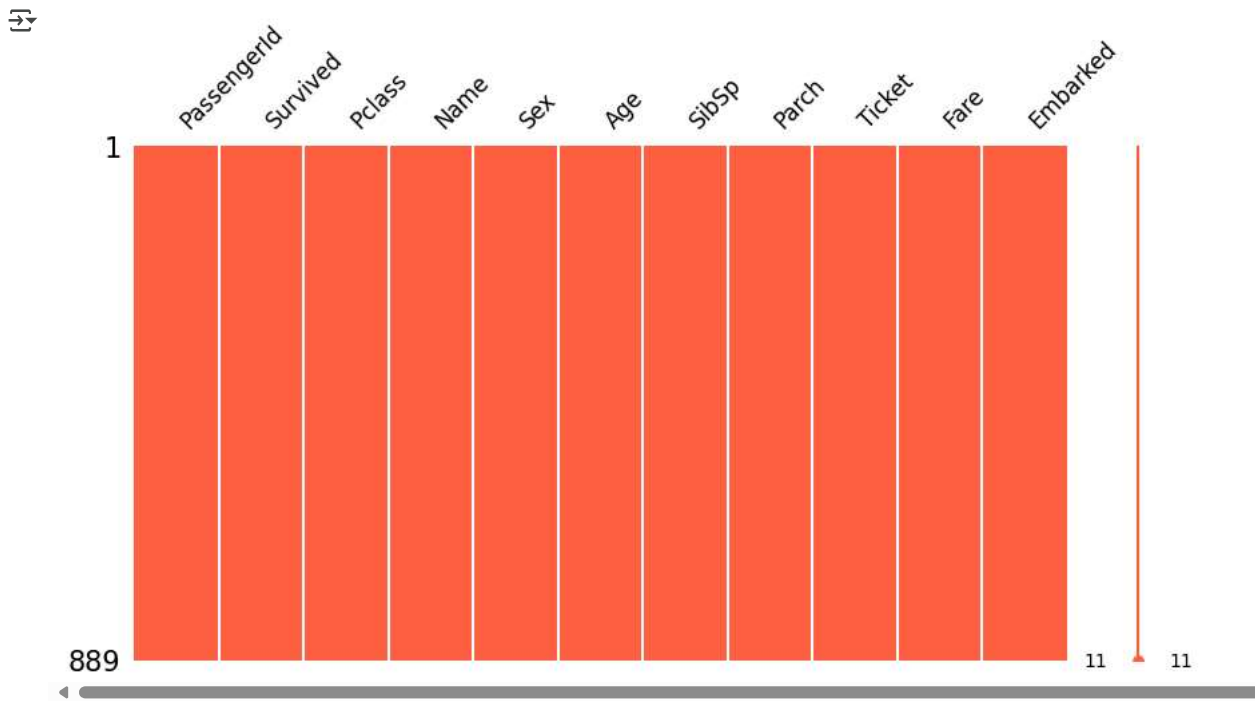
```
<ipython-input-34-81a1c153b904>:1: FutureWarning: Series.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.  
dat['Embarked'] = dat['Embarked'].fillna(method='bfill')
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
1											

```
import missingno  
missingno.bar(dat, color="dodgerblue", sort="ascending", figsize=(10,5), fontsize=12);
```

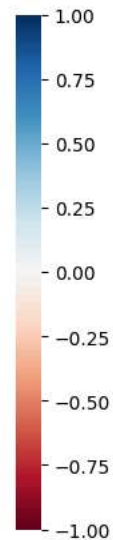


```
missingno.matrix(dat, figsize=(10,5), fontsize=12, color=(1, 0.38, 0.27));
```

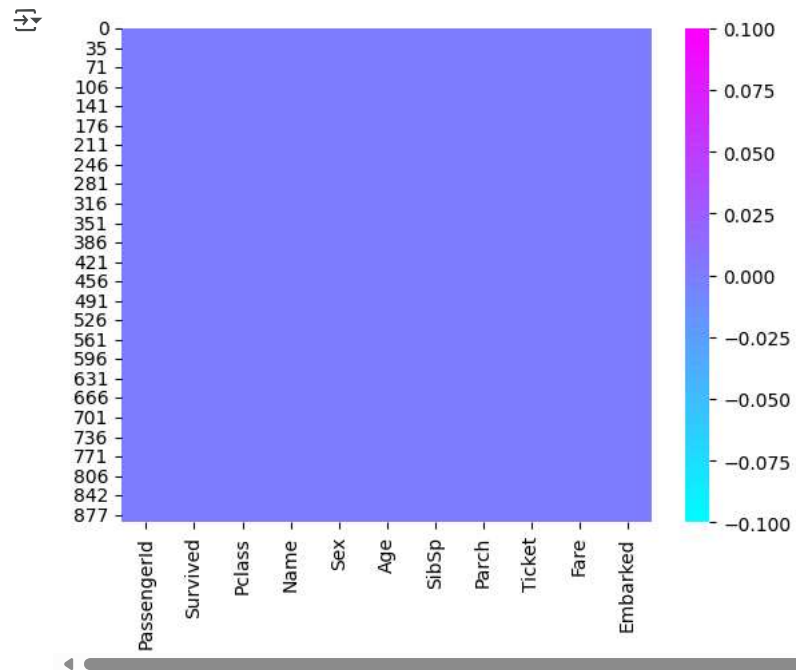


```
missingno.heatmap(dat, figsize=(10,5), fontsize=12);
```


```
/usr/local/lib/python3.11/dist-packages/seaborn/matrix.py:309: UserWarning: Attempting to set identical low and high xlims makes transformation singular; automatically expanding.  
ax.set(xlim=(0, self.data.shape[1]), ylim=(0, self.data.shape[0]))  
/usr/local/lib/python3.11/dist-packages/seaborn/matrix.py:309: UserWarning: Attempting to set identical low and high ylims makes transformation singular; automatically expanding.  
ax.set(xlim=(0, self.data.shape[1]), ylim=(0, self.data.shape[0]))
```



```
sns.heatmap(dat.isnull(),cmap='cool');
```



```
X = dat.iloc[:, :-1]
Y = dat.iloc[:, -1]
Y.head(2)
```




	Embarked
0	2
1	0

dtype: int64


```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)
```

```
X_train.head()
```




	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
708	709	1	1	163	0	22.0	0	0	36	151.5500
240	241	0	3	881	0	28.0	1	0	192	14.4542
382	383	0	3	809	1	32.0	0	0	664	7.9250
792	793	0	3	718	0	28.0	8	2	567	69.5500
683	684	0	3	301	1	14.0	5	2	565	46.9000

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, Y_train)
```




▼ LinearRegression ⓘ ?  
 LinearRegression()

```
model.coef_
```




```
array([[-1.20964640e-05, -1.77753124e-01,  1.09279737e-03, -7.29354383e-05,
         5.15782516e-02,  2.74665729e-03,  6.12546040e-02,  8.63680910e-02,
        -1.26291446e-04, -3.44605501e-03]])
```

```
model.intercept_
```



```
1.6117612607154324
```


```
yp = model.predict(X_test)
yp
```



```
array([ 1.62552488,  1.27427317,  1.431521  ,  1.63242546,  1.31752724,
        1.46109008,  1.61315177,  1.60414183,  1.23177011,  1.03327939,
        1.57258952,  1.40060834,  1.69534187,  1.71120072,  1.60777251,
        1.54700347,  1.39607248,  1.62992899,  1.64627241,  1.63512729,
        1.88226915,  1.39183527,  1.73023687,  1.53103423,  1.60252433,
        1.54275554,  1.6017065  ,  1.42208112,  1.55781533,  1.64108038,
        1.3800597  ,  1.38782151,  1.57685361,  1.55831248,  1.98498042,
        1.61683091,  1.55971494, -0.30293285,  1.62319386,  1.60612222,
        1.63089871,  1.5965571  ,  1.60044503,  1.64723392,  1.42406212,
        1.47817463,  1.68595956,  1.6238626  ,  1.67642481,  0.72327004,
        1.43435676,  1.41082631,  1.61434323,  1.70330281,  1.42682034,
        1.32405192,  1.73961727,  1.6557767  ,  1.56859599,  1.59880225,
        1.84057229,  1.72869927,  1.33541019,  1.57083168,  1.36880048,
        1.63401268,  1.65201348,  1.50142956,  1.30361364,  1.18098347,
        1.63163175,  1.32069738,  1.77770729,  1.45349944,  1.42852228,
        1.37598076,  1.59782155,  1.59800241,  1.57117284,  1.59235612,
        1.63245492,  1.34120455,  1.53337888,  1.43950931,  1.69398269,
        1.66782377,  1.67475039,  1.62869697,  1.61698807,  1.61424076,
        1.3968834  ,  1.51440935,  1.58591875,  1.65176379,  1.49588906,
        1.57652487,  1.65033255,  1.6321529  ,  1.52565446,  1.61004925,
        1.61946666,  0.77211196,  1.79486026,  1.62943334,  1.5117882  ,
        1.66915356,  1.36967159,  1.30949789,  1.46252633,  1.59216929,
        1.59080576,  1.31707057,  1.82468626,  1.37613948,  1.68785058,
        1.75026926,  1.6921736  ,  1.71023193,  1.40503278,  1.66801667,
        1.49538918,  1.13324898,  1.68182328,  1.58296426,  1.21662538,
```

1.22421557, 1.51214173, 1.34061828, 1.57568179, 1.66154699,  
1.36574329, 1.54399206, 1.50995567, 1.68011144, 1.39166048,  
1.48601998, 1.64539563, 1.42533212, 1.61353236, 1.45430576,  
1.55121533, 1.61315187, 1.39895685, 1.93260742, 1.09773989,  
1.53942911, 1.67871505, 0.93755066, 1.62553754, 1.38136617,  
1.64339044, 1.45527373, 1.56416005, 1.53280612, 1.78596646,  
1.49428684, 1.56457224, 1.65123633, 1.72676876, 1.22104972,  
1.49921486, 1.03989354, 1.42859251, 1.63936681, 1.6662746 ,  
1.70595782, 1.34653905, 1.6167377 , 1.65030917, 1.50486869,  
1.54845803, 1.50878678, 1.6919405 , 1.57010093, 1.53754038,  
1.88164425, 1.62116943, 1.40759331]])

Y\_test

	Embarked
281	2
435	2
39	0
418	2
585	2
...	...
433	2
807	2
25	2
85	2
10	2