

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

```
%matplotlib inline
```

```
train = pd.read_csv('/titanic_train.csv')
```

```
train.head()
```

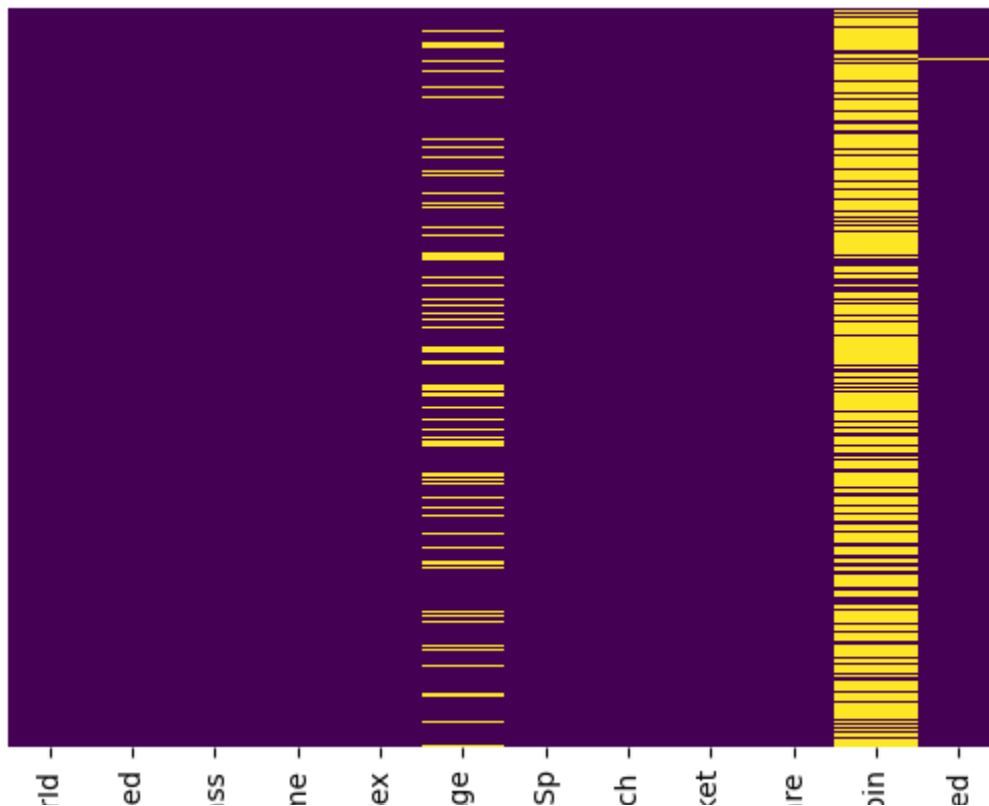
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs)	female	38.0	1	0	PC 17599	71.28

Next steps:

[View recommended plots](#)

```
sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis') #To get the columns which has n
```

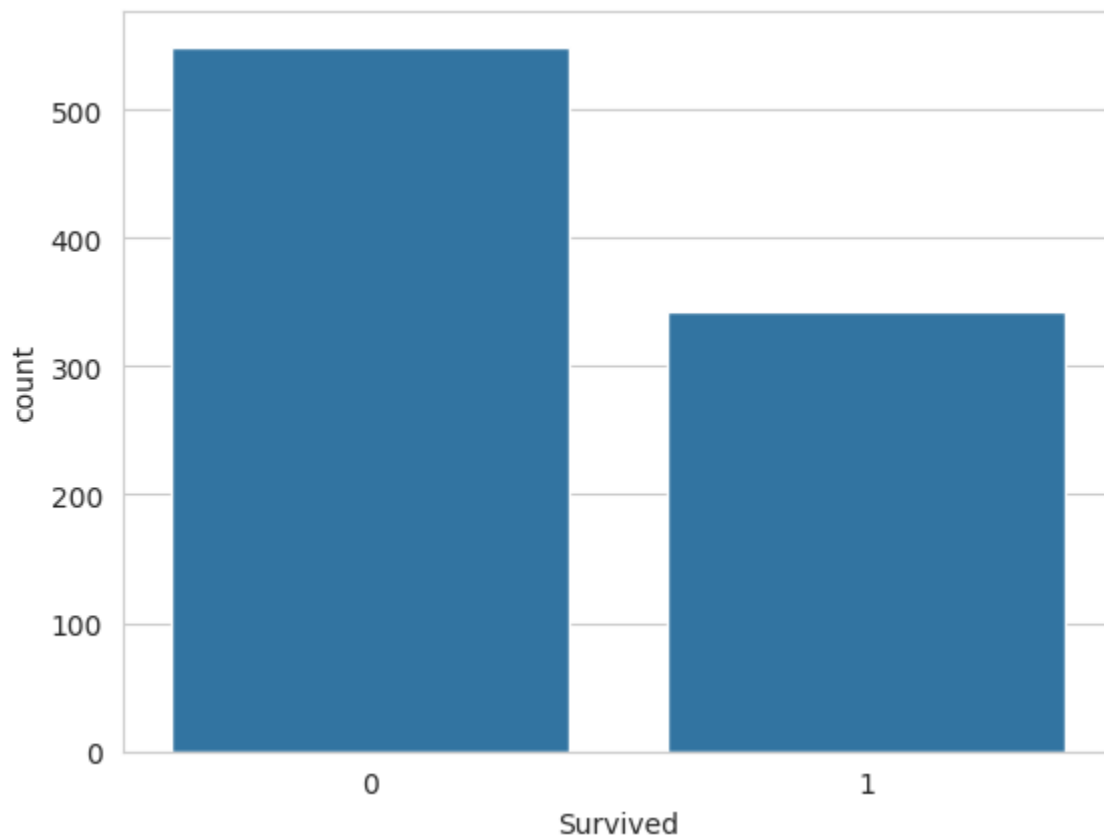
<Axes: >



PassengerId
Survived
Pclass
Name
Sex
Age
Siblings
Parthers
Ticket
Fare
Cabin
Embarked

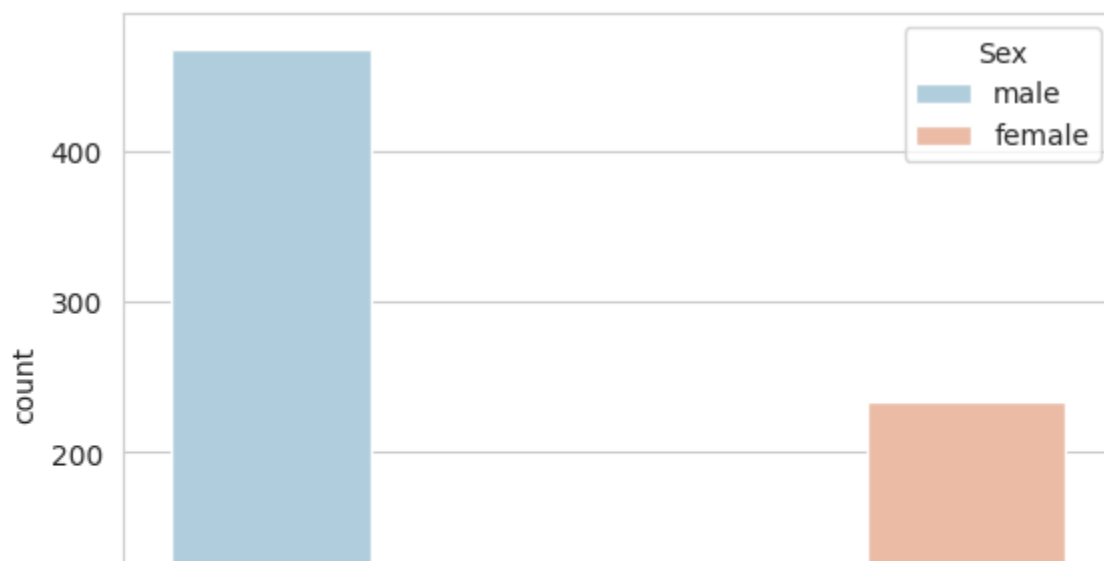
```
sns.set_style('whitegrid')  
sns.countplot(x='Survived',data=train)
```

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



```
sns.countplot(x='Survived',hue='Sex',data=train,palette='RdBu_r') #survival based on sex
```

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



```
sns.countplot(x='Survived',hue='Pclass',data=train) #survival based on passenger class
```

```
sns.distplot(train['Age'].dropna(),kde=False,bins=30)
```

```
sns.countplot(x='SibSp',data=train) # count based on sibling or spouse
```

```
train['Fare'].hist(bins=40,figsize=(10,4))
```

```
import cufflinks as cf  
cf.go_offline()
```

```
train['Fare'].iplot(kind='hist',bins=40)
```

```
sns.countplot(x='Parch',data=train)
```

```
plt.figure(figsize=(10,7))  
sns.boxplot(x='Pclass',y='Age',data=train)
```

```
def inpute_age(cols):
    Age = cols[0]
    Pclass = cols[1]
    if pd.isnull(Age):
        if Pclass == 1:
            return 37
        elif Pclass == 2:
            return 29
        else:
            return 24
    else:
        return Age

train['Age'] = train[['Age', 'Pclass']].apply(inpute_age,axis=1)

sns.heatmap(train.isnull(),yticklabels=False,cbar=False)
```

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

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

```
sex = pd.get_dummies(train['Sex'],drop_first=True)
```

```
sex.head()
```

```
embark = pd.get_dummies(train['Embarked'],drop_first=True)
```

```
embark.head()
```

Next steps:

 [View recommended plots](#)


```
train = pd.concat([train,sex,embark],axis=1)
```

```
train.head()
```

```
train.head()
```

```
train.drop('PassengerId',axis=1,inplace=True)
```

```
train.head()
```

```
pclass = pd.get_dummies(train['Pclass'])
```

```
pclass.head()
```

Next steps: [View recommended plots](#)

Next steps: [View recommended plots](#)

```
X= train.drop('Survived',axis=1)
y= train['Survived'] #trying to predict
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

```
from sklearn.linear_model import LogisticRegression
```

```
logmodel = LogisticRegression()
```

```
logmodel.fit(X_train,y_train)
```

```
predictions = logmodel.predict(X_test)
```

```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

```
confusion_matrix(y_test,predictions)
```

```
array([[149,  14],
       [ 31,  73]])
```

