```
1 #import libraries
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 %matplotlib inline
```

```
1 #reading data
2 import pandas as pd
3 train=pd.read_csv("train.csv")
4 train
```

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

```
1 train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	7

```
1 #info
2 train.info()
```

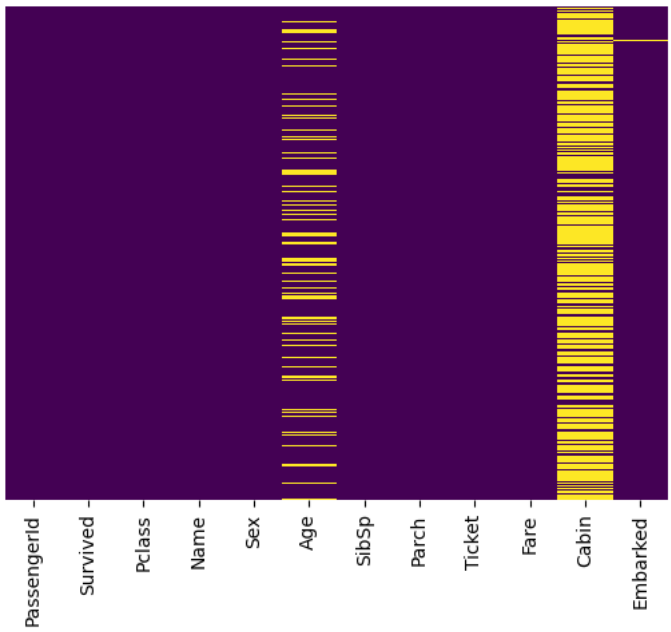
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    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
```

```
1 #we can  use seaborn to create a simple heatmap to see where the missing data
2 train.isnull()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False
887	False	False	False	False	False	False	False	False	False	False

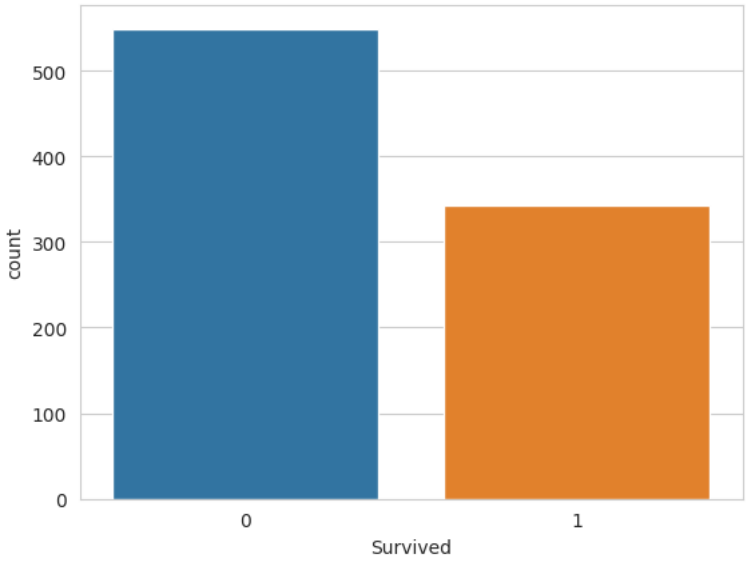
```
1 #visualisation
2 import seaborn as sns
3 sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<Axes: >
```



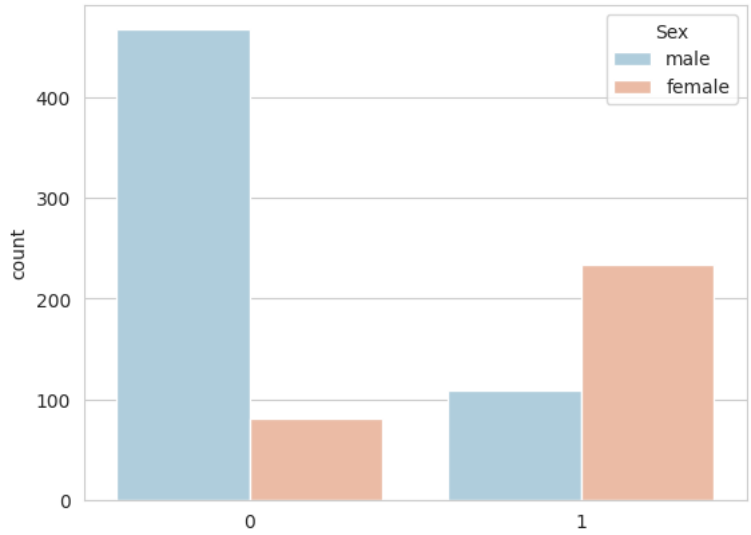

```
1 #graphical representation of data
2 sns.set_style('whitegrid')
3 sns.countplot(x='Survived',data=train)
```

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



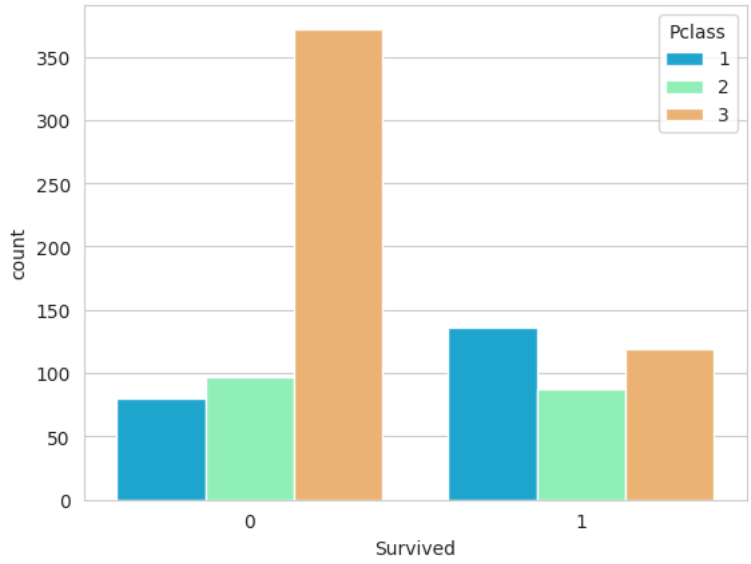

```
1 sns.set_style('whitegrid')
2 sns.countplot(x='Survived',hue='Sex',data=train,palette='RdBu_r')
```

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




```
1 sns.set_style('whitegrid')
2 sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')
```

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




```
1 sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=40)
```

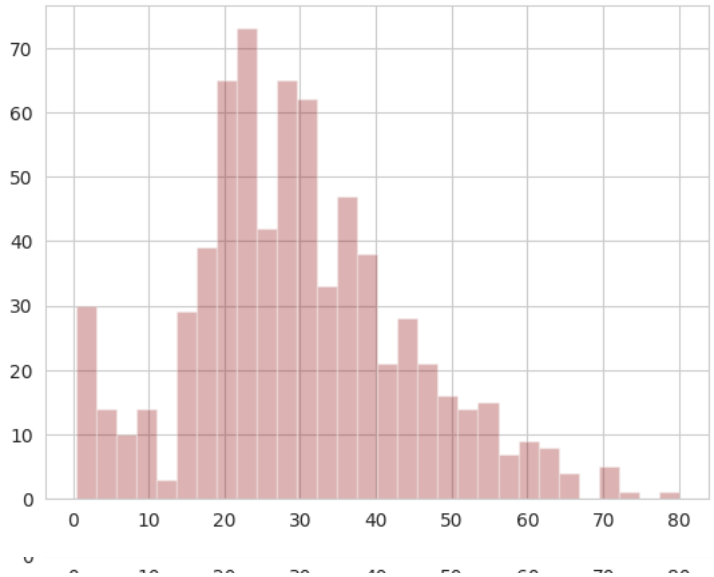
```
<ipython-input-22-ac571356fd3a>:1: UserWarning:

`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).
```

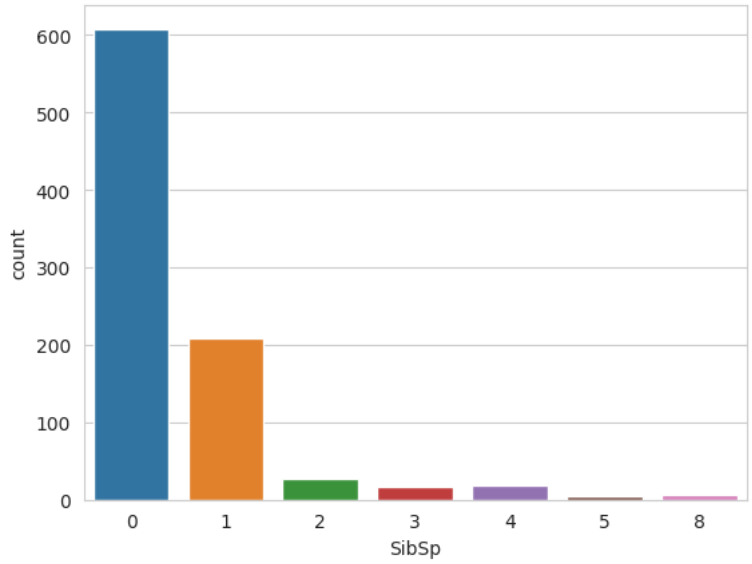
```
1 train['Age'].hist(bins=30,color='darkred',alpha=0.3)
```

```
<Axes: >
```



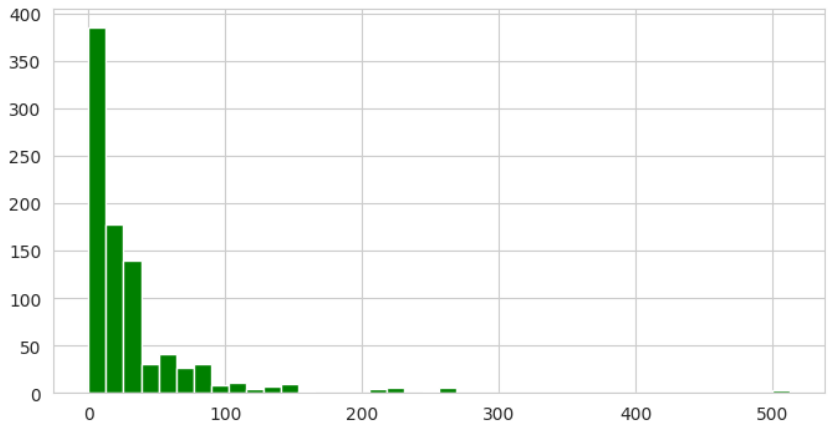

```
1 sns.countplot(x='SibSp',data=train)
```

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



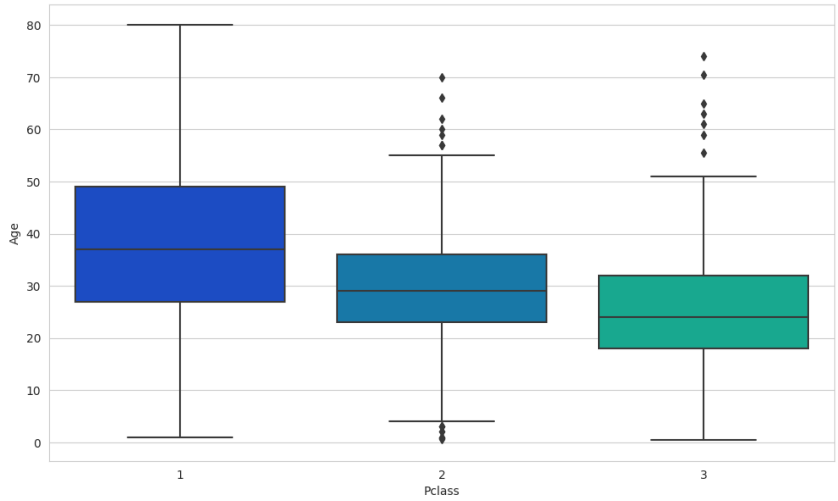

```
1 train['Fare'].hist(color='green',bins=40,figsize=(8,4))
```

```
<Axes: >
```



```
import matplotlib.pyplot as plt
plt.figure(figsize=(12,7))
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
```

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




```
#create a function
def impute_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(impute_Age,axis=1)

```

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

```
<Axes: >
```

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

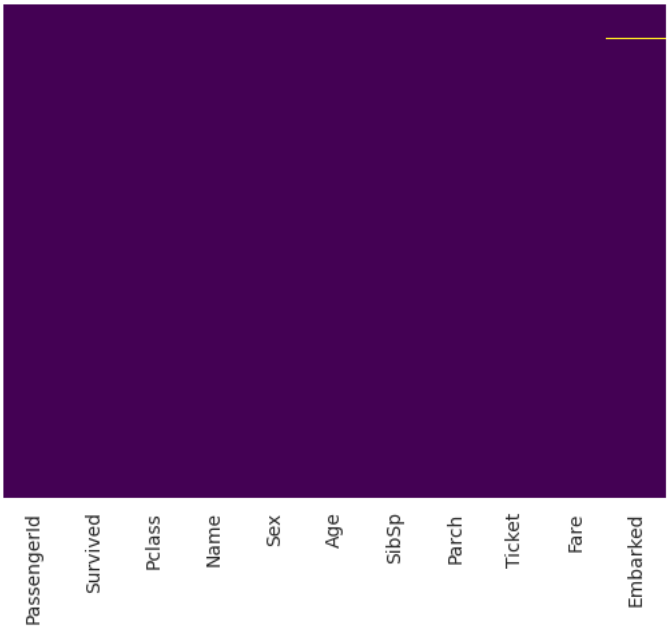


```
1 train.head()
```

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

```
1 sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<Axes: >
```




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

```
1 train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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    object 
 4   Sex          889 non-null    object 
 5   Age          889 non-null    float64
 6   SibSp        889 non-null    int64  
 7   Parch        889 non-null    int64  
 8   Ticket       889 non-null    object 
 9   Fare         889 non-null    float64
 10  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(4)
memory usage: 83.3+ KB
```

```
1 import pandas as pd
2 pd.get_dummies(train['Embarked'],drop_first=True).head()
```

	Q	S
0	0	1
1	0	0
2	0	1

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

```
1 train.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace=True)
```

```
1 train.head()
```

	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

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

```
1 train.head()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	1	0	3	22.0	1	0	7.2500	1	0	1
1	2	1	1	38.0	1	0	71.2833	0	0	0
2	3	1	3	26.0	0	0	7.9250	0	0	1
3	4	1	1	35.0	1	0	53.1000	0	0	1
4	5	0	3	35.0	0	0	8.0500	1	0	1

```
1 train.drop('Survived',axis=1).head()
2
```

	PassengerId	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	1	3	22.0	1	0	7.2500	1	0	1
1	2	1	38.0	1	0	71.2833	0	0	0
2	3	3	26.0	0	0	7.9250	0	0	1
3	4	1	35.0	1	0	53.1000	0	0	1
4	5	3	35.0	0	0	8.0500	1	0	1

```
1 train['Survived'].head()
```

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

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

```
1 X_train,X_test,Y_train,Y_test =train_test_split(train.drop('Survived',axis=1),train['Survived'],test_size=0.30,random_state=101)
```

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

```
1 logmodel=LogisticRegression()
2 logmodel.fit(X_train,Y_train)
```

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

```
1 from sklearn.metrics import confusion_matrix
```

```
1 accuracy=confusion_matrix(Y_test,predictions)
```

```
1 "accuracy is:",accuracy
```

```
('accuracy is:', 0.8014981273408239)
```

```
1 from sklearn.metrics import accuracy_score
```

```
1 #accuracy
2 accuracy=accuracy_score(Y_test,predictions)
3 accuracy
```

```
0.8014981273408239
```

```
1 #prediction
2 predictions
```

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

```
1 #major project completed successfully
2 #acuuracy is 0.8014981273408239
3 #Done by MANIKANDAN M
4 #Thank you..
```