

```
In [1]: import warnings
warnings.filterwarnings("ignore")

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

```
In [2]: df = pd.read_csv("titanic_train.csv")
df.head(20)
```

Out[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
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
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
12	13	0	3	Saunderscock, Mr. William Henry	male	20.0	0	0	A/5. 2151	8.0500	NaN	S
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2750	NaN	S
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	350406	7.8542	NaN	S
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	0	0	248706	16.0000	NaN	S
16	17	0	3	Rice, Master. Eugene	male	2.0	4	1	382652	29.1250	NaN	Q
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0000	NaN	S
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0	1	0	345763	18.0000	NaN	S
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN	C

```
In [3]: df["Name"].to_list()
```

```
Out[3]: ['Braund, Mr. Owen Harris',
'Cumings, Mrs. John Bradley (Florence Briggs Thayer)',
'Heikkinen, Miss. Laina',
'Futrelle, Mrs. Jacques Heath (Lily May Peel)',
'Allen, Mr. William Henry',
'Moran, Mr. James',
'McCarthy, Mr. Timothy J',
'Palsson, Master. Gosta Leonard',
'Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)',
'Nasser, Mrs. Nicholas (Adele Achem)',
'Sandstrom, Miss. Marguerite Rut',
'Bonnell, Miss. Elizabeth',
'Saunderscock, Mr. William Henry',
'Andersson, Mr. Anders Johan',
'Vestrom, Miss. Hulda Amanda Adolfina',
'Hewlett, Mrs. (Mary D Kingcome) ',
'Rice, Master. Eugene',
'Williams, Mr. Charles Eugene',
'Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)',
'Masselmani, Mrs. Fatima']
```

```
In [4]: df["Name"].unique()
```

```
Out[4]: array(['Braund, Mr. Owen Harris',
               'Cumings, Mrs. John Bradley (Florence Briggs Thayer)',
               'Heikkinen, Miss. Laina',
               'Futrelle, Mrs. Jacques Heath (Lily May Peel)',
               'Allen, Mr. William Henry', 'Moran, Mr. James',
               'McCarthy, Mr. Timothy J', 'Palsson, Master. Gosta Leonard',
               'Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)',
               'Nasser, Mrs. Nicholas (Adele Achem)',
               'Sandstrom, Miss. Marguerite Rut', 'Bonnell, Miss. Elizabeth',
               'Saunderscock, Mr. William Henry', 'Andersson, Mr. Anders Johan',
               'Vestrom, Miss. Hulda Amanda Adolfina',
               'Hewlett, Mrs. (Mary D Kingcome) ', 'Rice, Master. Eugene',
               'Williams, Mr. Charles Eugene',
               'Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)',
               'Masselmani, Mrs. Fatima', 'Fynney, Mr. Joseph J',
               'Beesley, Mr. Lawrence', 'McGowan, Miss. Anna "Annie"',
               'Sloper, Mr. William Thompson', 'Palsson, Miss. Torborg Danira',
               'Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)',
               'Emir, Mr. Farred Chehab', 'Fortune, Mr. Charles Alexander',
               ...])
```

**Goal:- Given the features we need to predict if passenger will survived or not**

## EDA

```
In [5]: df.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
```

```
In [6]: df.isna().sum()
```

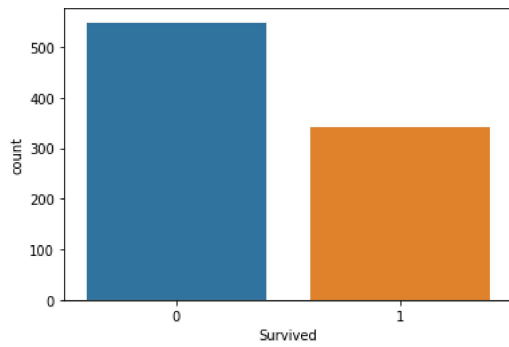
```
Out[6]: PassengerId    0
Survived              0
Pclass                0
Name                  0
Sex                   0
Age                  177
SibSp                 0
Parch                 0
Ticket                0
Fare                  0
Cabin                 687
Embarked              2
dtype: int64
```

```
In [7]: df["Survived"].value_counts()
```

```
Out[7]: 0    549
        1    342
        Name: Survived, dtype: int64
```

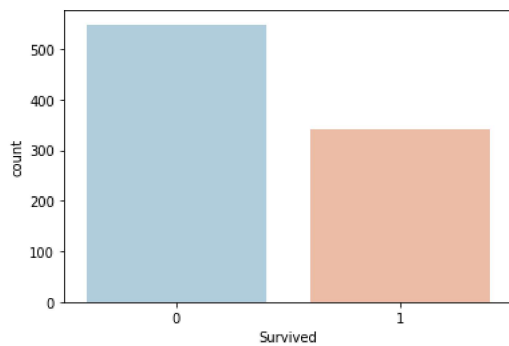
```
In [8]: sns.countplot(data=df, x="Survived")
```

```
Out[8]: <AxesSubplot: xlabel='Survived', ylabel='count'>
```

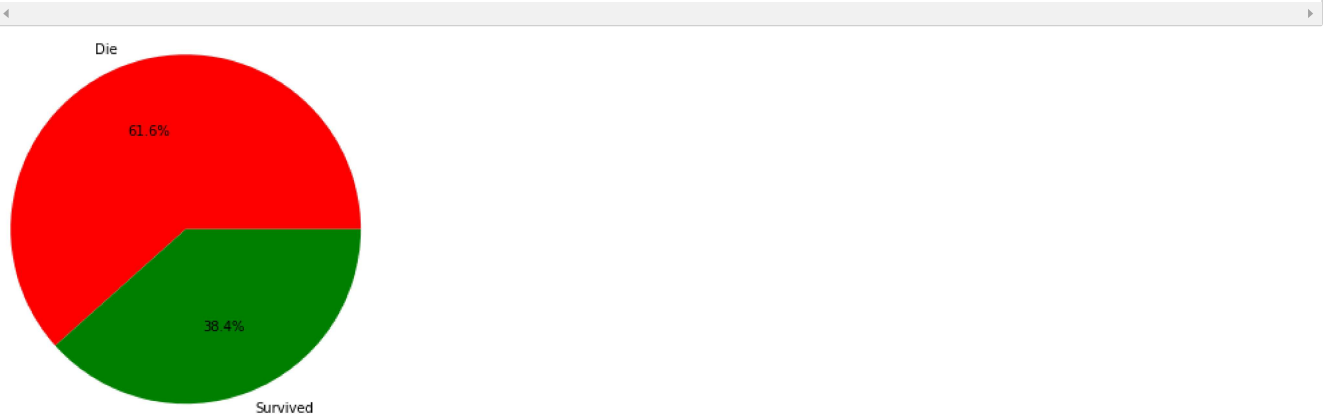


```
In [9]: sns.countplot(data=df, x="Survived", palette="RdBu_r")
```

```
Out[9]: <AxesSubplot: xlabel='Survived', ylabel='count'>
```



```
In [10]: plt.pie(df["Survived"].value_counts(), labels=["Die", "Survived"], colors=["red", "green"], autopct="%1.1f%%", radius=1.5)  
plt.show()
```

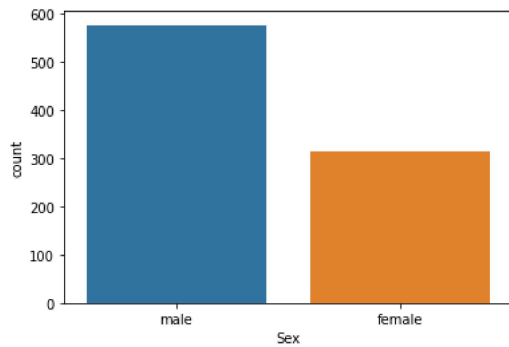


```
In [11]: df["Sex"].value_counts()
```

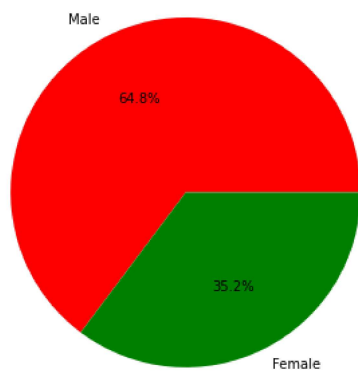
```
Out[11]: male      577  
female    314  
Name: Sex, dtype: int64
```

```
In [12]: sns.countplot(data=df,x="Sex")
```

```
Out[12]: <AxesSubplot: xlabel='Sex', ylabel='count'>
```

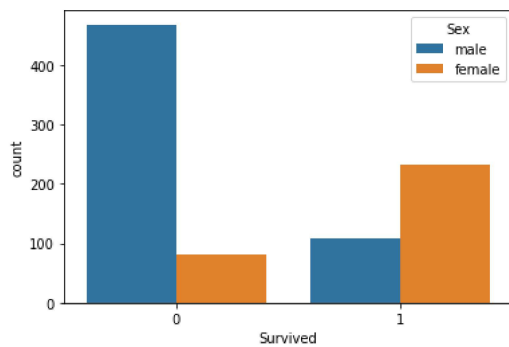


```
In [13]: plt.pie(df["Sex"].value_counts(), labels=["Male","Female"], colors=["red","green"], autopct="%1.1f%%",radius=1.5)  
plt.show()
```



```
In [14]: sns.countplot(data=df,x="Survived",hue="Sex")
```

```
Out[14]: <AxesSubplot: xlabel='Survived', ylabel='count'>
```

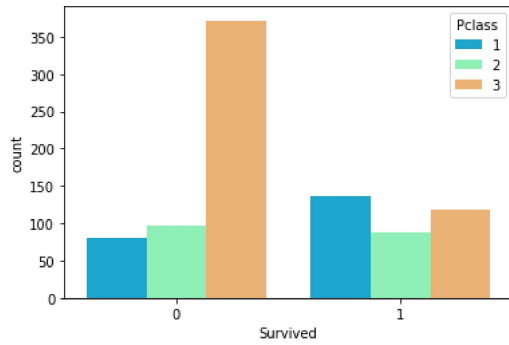


```
In [15]: df["Pclass"].value_counts()
```

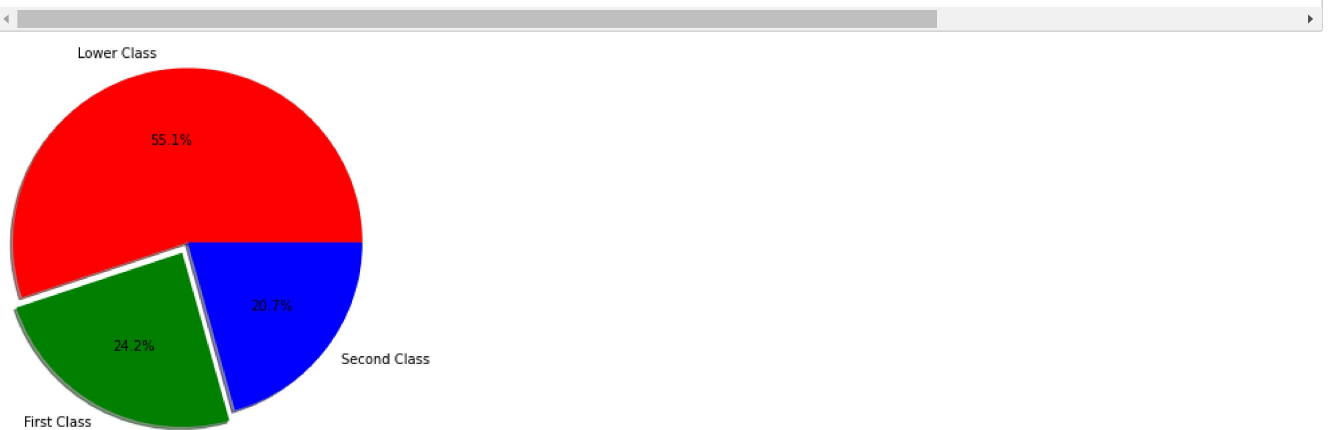
```
Out[15]: 3    491  
         1    216  
         2    184  
         Name: Pclass, dtype: int64
```

```
In [16]: sns.countplot(data=df, x="Survived", hue="Pclass", palette="rainbow")
```

```
Out[16]: <AxesSubplot: xlabel='Survived', ylabel='count'>
```



```
In [17]: plt.pie(df["Pclass"].value_counts(),explode=(0,0.1,0),shadow=True,labels=["Lower Class","First Class","Second Class"], colors=["r","g","b"], autopct=True)
```

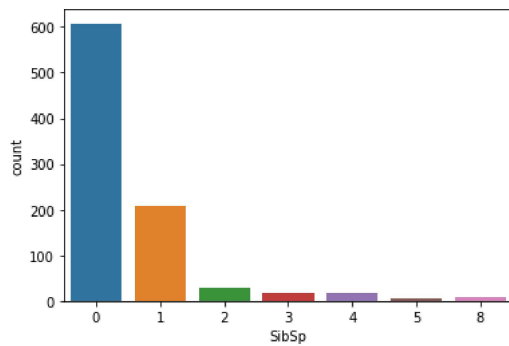


```
In [18]: df["SibSp"].value_counts()
```

```
Out[18]: 0    608
         1    209
         2     28
         4     18
         3     16
         8      7
         5      5
         Name: SibSp, dtype: int64
```

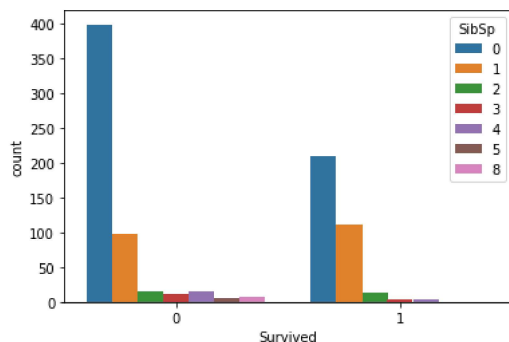
```
In [19]: sns.countplot(data=df, x="SibSp")
```

```
Out[19]: <AxesSubplot: xlabel='SibSp', ylabel='count'>
```



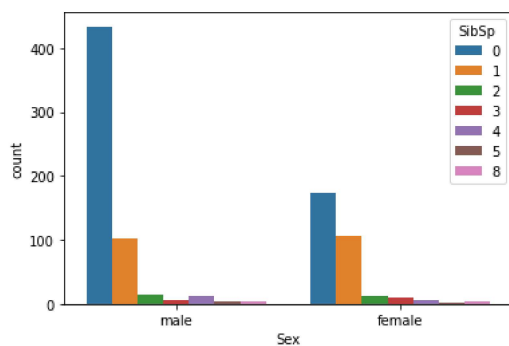
```
In [20]: sns.countplot(data=df,x="Survived",hue="SibSp")
```

```
Out[20]: <AxesSubplot: xlabel='Survived', ylabel='count'>
```



```
In [21]: sns.countplot(data=df,x="Sex",hue="SibSp")
```

```
Out[21]: <AxesSubplot: xlabel='Sex', ylabel='count'>
```

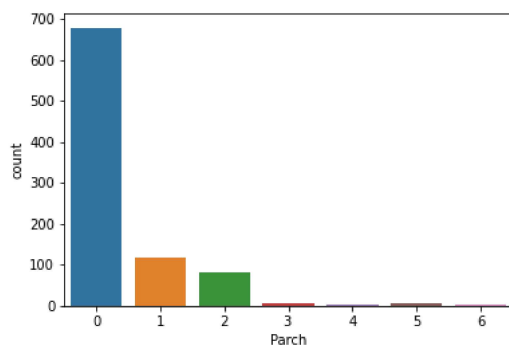


```
In [22]: df["Parch"].value_counts()
```

```
Out[22]: 0    678
         1    118
         2     80
         5      5
         3      5
         4      4
         6      1
         Name: Parch, dtype: int64
```

```
In [23]: sns.countplot(data=df,x="Parch")
```

```
Out[23]: <AxesSubplot: xlabel='Parch', ylabel='count'>
```



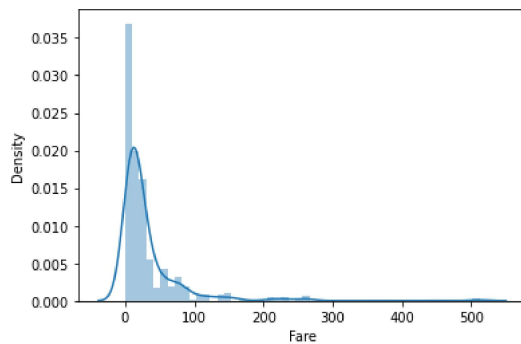
```
In [24]: df.describe()
```

```
Out[24]:
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

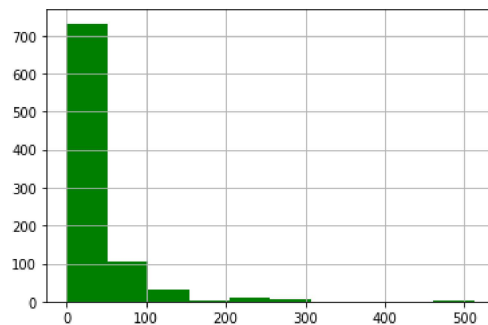
```
In [25]: sns.distplot(df["Fare"])
```

```
Out[25]: <AxesSubplot: xlabel='Fare', ylabel='Density'>
```

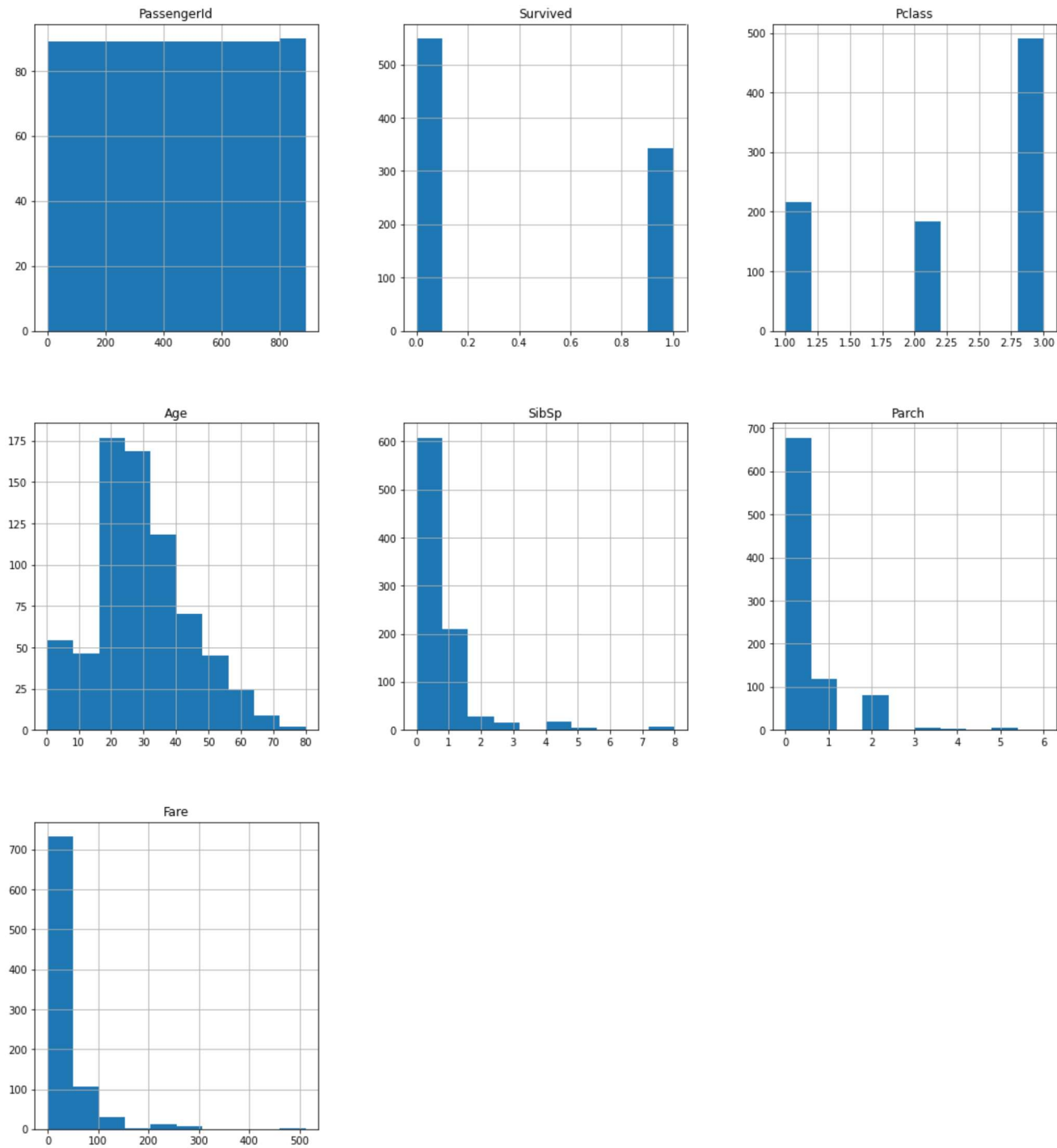


```
In [26]: df["Fare"].hist(color="green")
```

```
Out[26]: <AxesSubplot: >
```



```
In [27]: df.hist(figsize=(18,20))
plt.show()
```





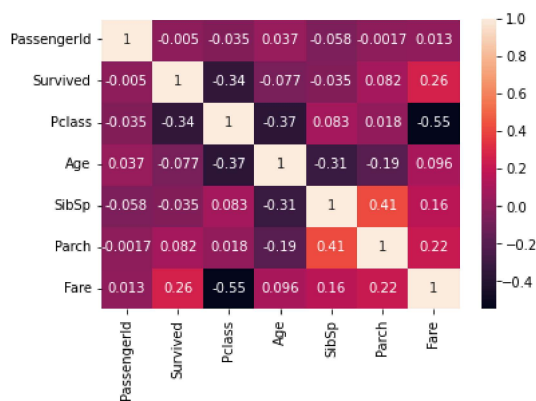
```
In [28]: sns.pairplot(df,hue ="Survived")
```

```
Out[28]: <seaborn.axisgrid.PairGrid at 0x207ff4056f0>
```



```
In [29]: sns.heatmap(df.corr(),annot=True)
```

```
Out[29]: <AxesSubplot: >
```



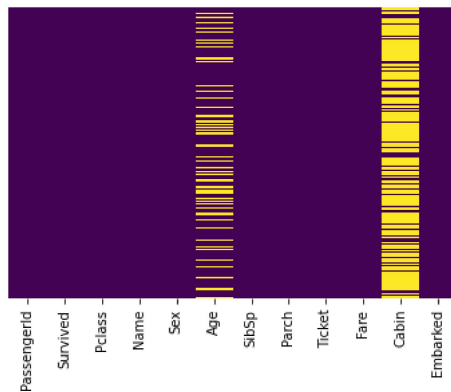
## Data Cleaning And Preprocessing

In [31]: `df.isna().sum()`

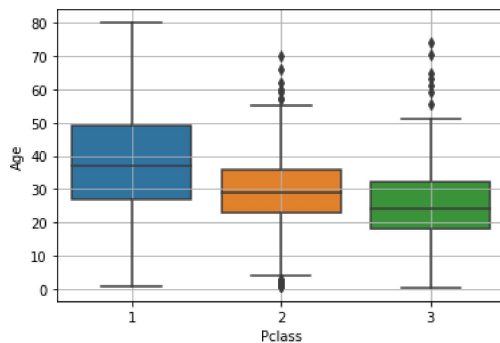
```
Out[31]: PassengerId    0
Survived      0
Pclass        0
Name          0
Sex           0
Age          177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin        687
Embarked      2
dtype: int64
```

In [32]: `sns.heatmap(df.isnull(),yticklabels=False, cbar=False, cmap="viridis")`

Out[32]: <AxesSubplot: >



In [33]: `sns.boxplot(x="Pclass",y="Age",data=df)  
plt.grid()`



In [34]: `def fillage(col):`

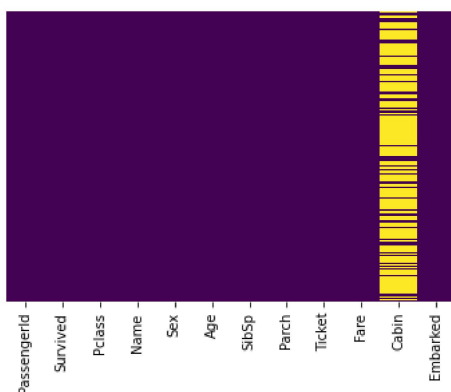
```
    age=col[0]
    pclass=col[1]

    if(pd.isnull(age)):
        if(pclass==1):
            return 38
        elif pclass==2:
            return 29
        else:
            return 25
    else:
        return age
```

In [35]: `df["Age"]=df[["Age", "Pclass"]].apply(fillage,axis=1)`

```
In [36]: sns.heatmap(df.isnull(),yticklabels=False, cbar=False, cmap="viridis")
```

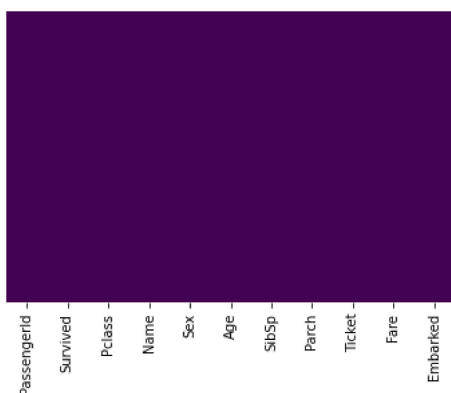
```
Out[36]: <AxesSubplot: >
```



```
In [37]: df.drop("Cabin",axis=1,inplace=True)
```

```
In [38]: sns.heatmap(df.isnull(),yticklabels=False, cbar=False, cmap="viridis")
```

```
Out[38]: <AxesSubplot: >
```



```
In [39]: df.isnull().sum()
```

```
Out[39]: PassengerId    0
Survived              0
Pclass                0
Name                  0
Sex                   0
Age                   0
SibSp                 0
Parch                 0
Ticket                0
Fare                  0
Embarked              2
dtype: int64
```

```
In [40]: df.dropna(inplace=True)
```

```
In [41]: df.isnull().sum()
```

```
Out[41]: PassengerId    0
Survived              0
Pclass                0
Name                  0
Sex                   0
Age                   0
SibSp                 0
Parch                 0
Ticket                0
Fare                  0
Embarked              0
dtype: int64
```

```
In [42]: df.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
```

```
In [43]: df.head()
```

Out[43]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S

```
In [44]: df.drop(["PassengerId", "Name", "Ticket"],axis=1,inplace=True)
```

```
In [45]: df.head()
```

Out[45]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S

## Data Preprocessing

```
In [46]: x = df.iloc[:,1:]
```

```
In [47]: x
```

...

```
In [48]: y = df.iloc[:,0]
```

```
In [49]: y
```

...

```
In [50]: x.Sex.value_counts()
```

Out[50]: male 577  
female 312  
Name: Sex, dtype: int64

```
In [51]: x.Embarked.value_counts()
```

Out[51]: S 644  
C 168  
Q 77  
Name: Embarked, dtype: int64

```
In [52]: from sklearn.preprocessing import OrdinalEncoder
oe = OrdinalEncoder()

x[["Sex", "Embarked"]]=oe.fit_transform(x[["Sex", "Embarked"]])
```

```
In [53]: x
```

```
Out[53]:
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1.0	22.0	1	0	7.2500	2.0
1	1	0.0	38.0	1	0	71.2833	0.0
2	3	0.0	26.0	0	0	7.9250	2.0
3	1	0.0	35.0	1	0	53.1000	2.0
4	3	1.0	35.0	0	0	8.0500	2.0
...	...	...	...	...	...	...	...
886	2	1.0	27.0	0	0	13.0000	2.0
887	1	0.0	19.0	0	0	30.0000	2.0
888	3	0.0	25.0	1	2	23.4500	2.0
889	1	1.0	26.0	0	0	30.0000	0.0
890	3	1.0	32.0	0	0	7.7500	1.0

889 rows × 7 columns

```
In [54]: x.Sex.value_counts()
```

```
Out[54]: 1.0    577
0.0    312
Name: Sex, dtype: int64
```

```
In [55]: x.Embarked.value_counts()
```

```
Out[55]: 2.0    644
0.0    168
1.0     77
Name: Embarked, dtype: int64
```

## Model Building

```
In [56]: from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

```
In [57]: from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression()

logreg.fit(xtrain,ytrain)

ypred = logreg.predict(xtest)
```

```
In [60]: from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

ac = accuracy_score(ytest,ypred)
cm = confusion_matrix(ytest,ypred)
cr = classification_report(ytest,ypred)

print(f"Accuracy:- {ac}\n{cm}\n{cr}")
```

```
Accuracy:- 0.8426966292134831
[[144  22]
 [ 20  81]]
      precision    recall  f1-score   support

     0       0.88      0.87      0.87       166
     1       0.79      0.80      0.79       101

   accuracy                   0.84       267
  macro avg       0.83      0.83      0.83       267
 weighted avg       0.84      0.84      0.84       267
```

```
In [62]: train = logreg.score(xtrain,ytrain)
test = logreg.score(xtest,ytest)

print(f"Training Score :- {train}\n Testing Score:- {test}")
```

```
Training Score :- 0.7909967845659164
Testing Score:- 0.8426966292134831
```

```
In [63]: #High Bias + Low Variance ==>underfitting
```

```
In [64]: df.Survived.value_counts()
```

```
Out[64]: 0    549
         1    340
         Name: Survived, dtype: int64
```

```
In [65]: xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1,stratify=y)
```

```
In [66]: from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression()

logreg.fit(xtrain,ytrain)

ypred = logreg.predict(xtest)
```

```
In [67]: from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

ac = accuracy_score(ytest,ypred)
cm = confusion_matrix(ytest,ypred)
cr = classification_report(ytest,ypred)

print(f"Accuracy:- {ac}\n{cm}\n{cr}")
```

```
Accuracy:- 0.8014981273408239
[[140  25]
 [ 28  74]]
      precision    recall  f1-score   support

      0       0.83      0.85      0.84       165
      1       0.75      0.73      0.74       102

   accuracy          0.80          267
  macro avg       0.79      0.79      0.79       267
 weighted avg       0.80      0.80      0.80       267
```

```
In [68]: train = logreg.score(xtrain,ytrain)
test = logreg.score(xtest,ytest)

print(f"Training Score :- {train}\n Testing Score:- {test}")
```

```
Training Score :- 0.8038585209003215
Testing Score:- 0.8014981273408239
```

## Forecast New Observation

In [69]: x

Out[69]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1.0	22.0	1	0	7.2500	2.0
1	1	0.0	38.0	1	0	71.2833	0.0
2	3	0.0	26.0	0	0	7.9250	2.0
3	1	0.0	35.0	1	0	53.1000	2.0
4	3	1.0	35.0	0	0	8.0500	2.0
...	...	...	...	...	...	...	...
886	2	1.0	27.0	0	0	13.0000	2.0
887	1	0.0	19.0	0	0	30.0000	2.0
888	3	0.0	25.0	1	2	23.4500	2.0
889	1	1.0	26.0	0	0	30.0000	0.0
890	3	1.0	32.0	0	0	7.7500	1.0

889 rows × 7 columns

In [70]: oe

Out[70]:

```
OrdinalEncoder
OrdinalEncoder()
```

In [71]: oe.categories\_

Out[71]: [array(['female', 'male'], dtype=object), array(['C', 'Q', 'S'], dtype=object)]

In [72]:

```
s="male"
e="Q"

oe.transform([[s,e]])
```

Out[72]: array([[1., 1.]])

In [74]:

```
pclass=1
sex="male"
age=24
sibsp=0
parch=1
fare=34
embarked="S"

newob=[pclass,sex,age,sibsp,parch,fare,embarked]
newob[1]
newob[-1]
```

Out[74]: 'S'

In [81]: oe.transform([[newob[1],newob[-1]]])[0]

Out[81]: array([1., 2.])

In [83]: newob[1],newob[-1]=oe.transform([[newob[1],newob[-1]]])[0]

In [84]: newob

Out[84]: [1, 1.0, 24, 0, 1, 34, 2.0]

In [85]: logreg.predict([newob])

Out[85]: array([1], dtype=int64)

In [100]: `df[(df["Fare"]>500)]`

Out[100]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
258	1	1	female	35.0	0	0	512.3292	C
679	1	1	male	36.0	0	1	512.3292	C
737	1	1	male	35.0	0	0	512.3292	C

```
In [86]: def predictsurvived():
    pclass=int(input("Enter Passenger Class :- "))
    sex=input("Enter Gender of the Passenger :- ")
    age=float(input("Enter the age of the passenger :- "))
    sibsp=int(input("Enter no of SibSp of the Passenger:- "))
    parch=int(input("Enter no of parch of the Passenger:- "))
    fare=int(input("Enter Ticket Price:- "))
    embarked=input("Enter Embarked/Port of the Passenger:- ")

    newob=[pclass,sex,age,sibsp,parch,fare,embarked]
    newob[1],newob[-1]=oe.transform([[newob[1],newob[-1]]])[0]

    v=logreg.predict([newob])[0]

    if v==1:
        print(f"\n With the Given Feature the Passenger Will Survived")
        return v
    else:
        print(f"\n With the Given Feature the Passenger Will Not Survived")
        return v
```

In [110]: `predictsurvived()`

```
Enter Passenger Class :- 1
Enter Gender of the Passenger :- male
Enter the age of the passenger :- 20
Enter no of SibSp of the Passenger:- 0
Enter no of parch of the Passenger:- 0
Enter Ticket Price:- 512
Enter Embarked/Port of the Passenger:- C

With the Given Feature the Passenger Will Survived
```

Out[110]: 1

In [ ]:

In [ ]: