+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ···

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
import math as mt
```

[2] ✓ 6.2s

```
data=pd.read_csv("C:/Users/SANDY/Desktop/DATA ANALYSIS FILES - JULY 2024/PROJECT 12 - USING PYTHON [CASE STUDY 03] [Titanic - Machine Learning from Disaster]/train.csv")
```

[3] ✓ 0.0s

```
df=pd.DataFrame(data)
```

[4] ✓ 0.0s

1. Display Top 5 Rows of The Dataset

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

[5] ✓ 0.0s

	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

2. Check the Last 3 Rows of The Dataset

```
df.tail(3)
```

[6] ✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

3. Find Shape of Our Dataset (Number of Rows & Number of Columns)

```
df.shape
```

[7] ✓ 0.0s

```
(891, 12)
```

4. Get Information About Our Dataset Like Total Number Rows, Total Number of Columns, Datatypes of Each Column And Memory Requirement

```
df.info()
```

[8] ✓ 0.0s

```
<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
```

5. Get Overall Statistics About The Dataframe

```
df.describe()
```

[9] ✓ 0.0s

	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

6. Data Filtering

```
df.columns
df[["Name","Age"]]
```

[10] ✓ 0.0s

	Name	Age
0	Braund, Mr. Owen Harris	22.0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0
2	Heikkinen, Miss. Laina	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0
4	Allen, Mr. William Henry	35.0
...	...	...
886	Montvila, Rev. Juozas	27.0
887	Graham, Miss. Margaret Edith	19.0
888	Johnston, Miss. Catherine Helen "Carrie"	NaN
889	Behr, Mr. Karl Howell	26.0
890	Dooley, Mr. Patrick	32.0

891 rows × 2 columns

```
df.columns
df["Sex"].value_counts()
sum(df["Sex"]=="male")
df[df["Sex"]=="male"].head(2)
```

[11] ✓ 0.0s

	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	7.25	NaN	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05	NaN	S

```
df.columns
df["Survived"].value_counts()
sum(df["Survived"]==1)
df[df["Survived"]==1].head(2)
```

[12] ✓ 0.0s

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

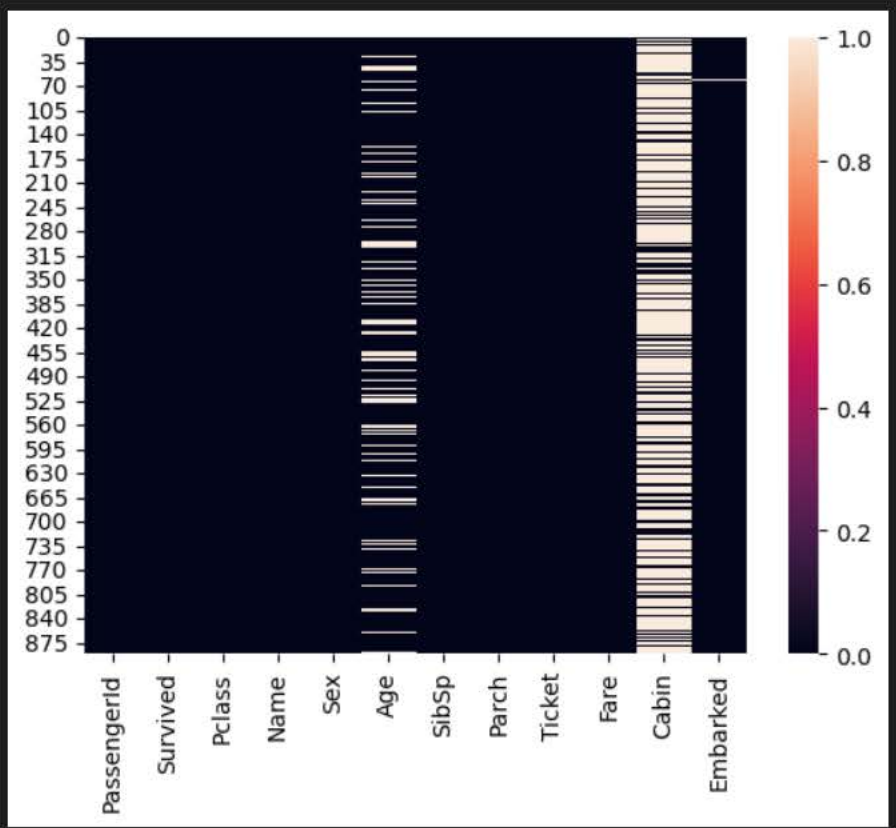
7.Check Null Values In The Dataset

```
[57] df.isnull().sum()
```

```
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
```

```
[58] sns.heatmap(df.isnull())
```

```
<Axes: >
```



PERCENTAGE OF NULL VALUES

PERCENTAGE OF NULL VALUES

```
[7]
perc_null = df.isnull().sum()*100/ len(df)
perc_null
```

```
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
```

8. Drop the Column

```
[8]
df.drop("Cabin", axis=1, inplace=True)
```

9. Handle Missing Values

```
[9]
df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Embarked         2
dtype: int64
```

```
[63]
df["Embarked"].mode()
```

```
[10]
df["Embarked"].fillna("S", inplace=True)
```

```
C:\Users\SANDY\AppData\Local\Temp\ipykernel_13360\2591539637.py:1: FutureWarning: A value is trying to be set on
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on w

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)'


  df["Embarked"].fillna("S", inplace=True)
```

```
[11]
df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Embarked         0
dtype: int64
```

```
[12]
df["Age"].value_counts()
df["Age"].fillna(df["Age"].mean(), inplace=True)
```

```
C:\Users\SANDY\AppData\Local\Temp\ipykernel_13360\693680584.py:2: FutureWarning: A value is trying to be set on
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on w

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)'


  df["Age"].fillna(df["Age"].mean(), inplace=True)
```

```
[13]
df.isnull().sum()
```

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
```

10. Categorical Data Encoding

```
df.columns
df.head(2)
df["Sex"].replace({"male":1, "female":2}, inplace=True)
df.head(2)
#INSERT A COLUMN AT A SPECIFIC INDEX
#df.insert(5,"Embarked_copy", value="Embarked")
df.head(2)
#df.drop("Embarked_copy", axis=1, inplace=True)
```

[15]

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	2	38.0	1	0	PC 17599	71.2833	C

```
df.head(2)
df["Embarked"].unique()
```

[16]

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

We will use get_dummies to encode "Embarked" column which has 3 unique values

```
df2=pd.get_dummies(df, columns=["Embarked"], drop_first=False)
df2["Embarked_C"].replace({True:1, False:0}, inplace=True)
df2["Embarked_Q"].replace({True:1, False:0}, inplace=True)
df2["Embarked_S"].replace({True:1, False:0}, inplace=True)
df2.head(2)
```

[17]

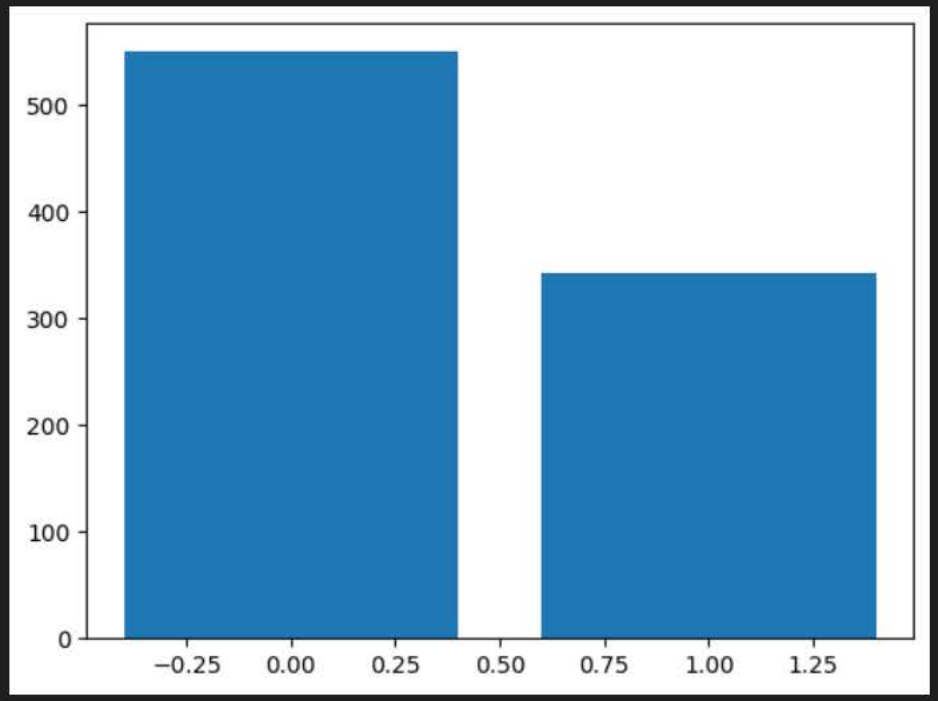
[18]
```
df2.drop("Embarked_C", axis=1, inplace=True)
```

[19]
```
df2.head(2)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_Q	Embarked_S
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	2	38.0	1	0	PC 17599	71.2833	0	0

11. What is Univariate Analysis?

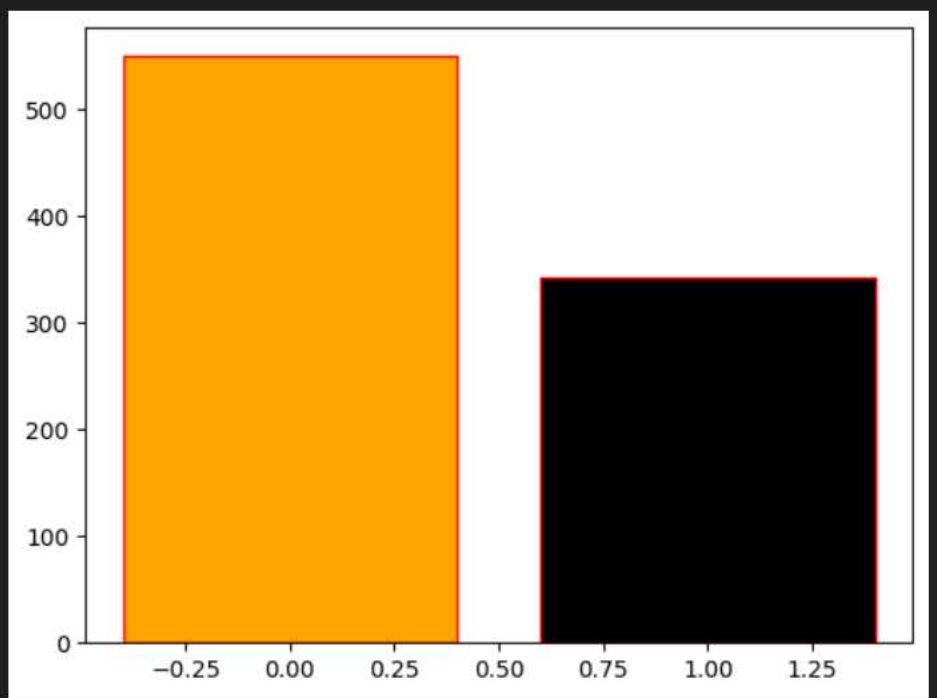
[23]
```
df2.columns
df3 = df2["Survived"].value_counts()
plt.bar(data=df3, x=df3.index, height=df3.values)
#sns.countplot(df3)
plt.show()
#survived = df2[df2["Survived"]==1].value_counts()
#died = df2[df2["Survived"]==0]["Survived"].value_counts()
#print("No of passengers that survived:", survived.count())
#print("No of passengers that died:", died.count())
```



How Many People Survived And How Many Died?

```
df4 = df2["Survived"].value_counts()
#df4
color=["orange", "black"]
plt.bar(data=df4, x=df4.index, height=df4.values, color=color, edgecolor="red")
plt.show()
```

[43]



How Many Passengers Were In First Class, Second Class, and Third Class?

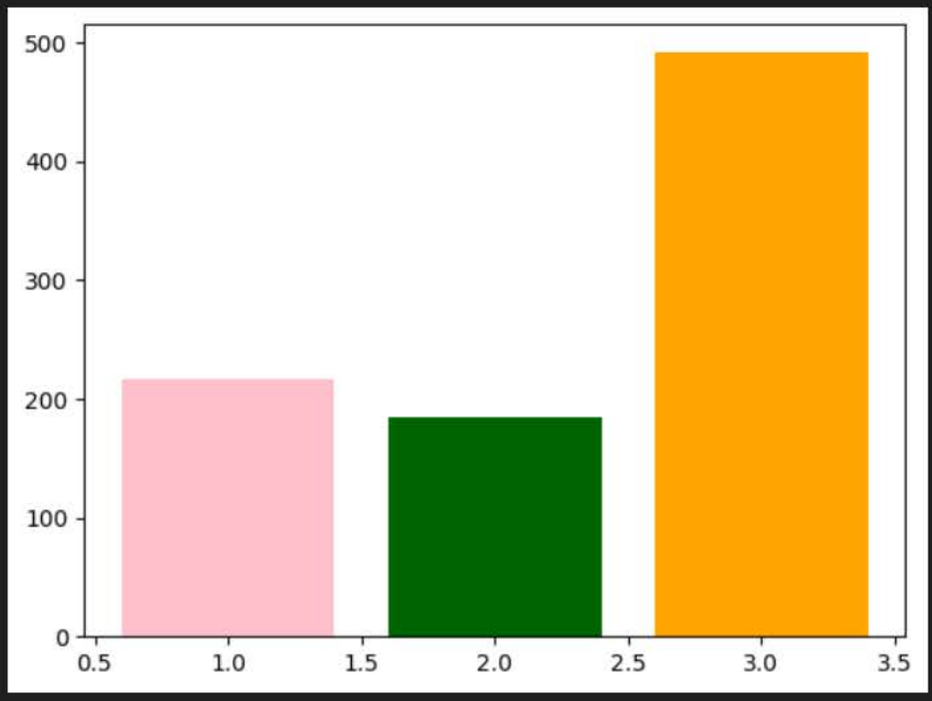
```
df2.columns
df5 = df2["Pclass"].value_counts()
df2["Pclass"].unique
first_class = df2[df2["Pclass"]==1]["Pclass"].count()
second_class = df2[df2["Pclass"]==2]["Pclass"].count()
third_class = df2[df2["Pclass"]==3]["Pclass"].count()
color=["orange","pink","darkgreen"]
plt.bar(data=df5, x=df5.index, height=df5.values, color=color)
plt.show()
#print("First Class Passengers: ", first_class)
#rint("Second Class Passengers: ", second_class)
#print("Third Class Passengers: ", third_class)
```

[53]

How Many Passengers Were In First Class, Second Class, and Third Class?

```
df2.columns
df5 = df2["Pclass"].value_counts()
df2["Pclass"].unique
first_class = df2[df2["Pclass"]==1]["Pclass"].count()
second_class = df2[df2["Pclass"]==2]["Pclass"].count()
third_class = df2[df2["Pclass"]==3]["Pclass"].count()
color=["orange","pink","darkgreen"]
plt.bar(data=df5, x=df5.index, height=df5.values, color=color)
plt.show()
#print("First Class Passengers: ", first_class)
#rint("Second Class Passengers: ", second_class)
#print("Third Class Passengers: ", third_class)
```

[53]



Number of Male And Female Passengers

```
df2.columns
df2["Sex"].value_counts()
```

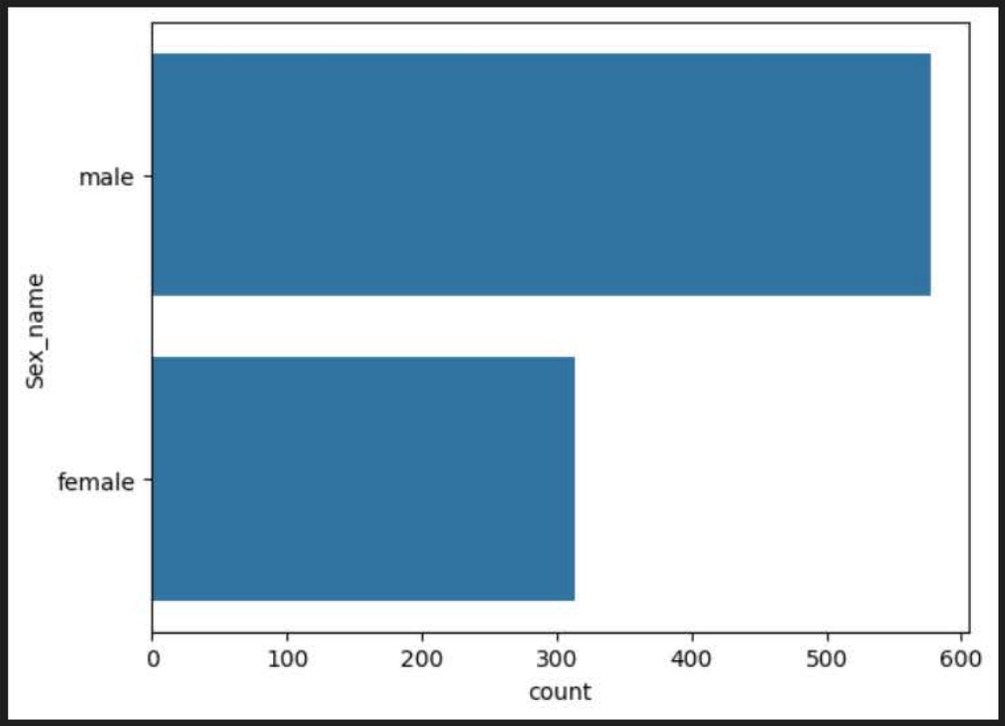
[55]

```
Sex
1    577
2    314
Name: count, dtype: int64
```

+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ...

```
df2["Sex_name"] = df2["Sex"].replace({1:"male",2:"female"})
df2.head(5)
sns.countplot(df2["Sex_name"])
plt.show()
```

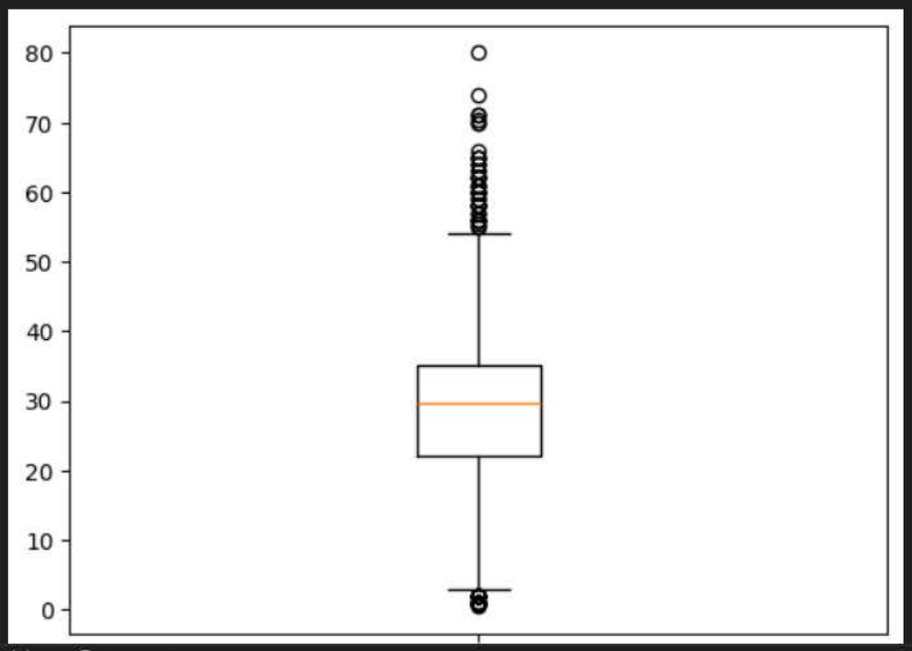
[60]



+ Code + Markdown

```
plt.boxplot(df2["Age"])
plt.show()
```

[65]



0 ⚠ 0 0 Connect

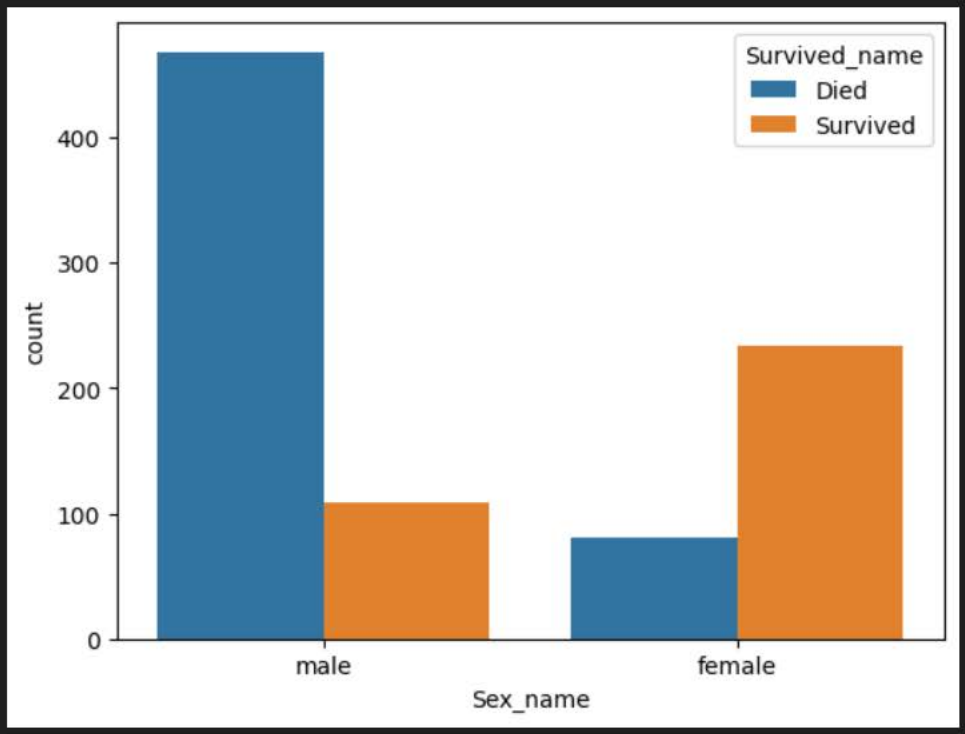
12. Bivariate Analysis

How Has Better Chance of Survival Male or Female?

```
df2.columns
df2["Survived_name"] = df2["Survived"].replace({0:"Died",1:"Survived"})
df2.head(2)
df2["Survived_name"] = df2["Survived"].replace({0:"Died", 1:"Survived"})

sns.countplot(data=df2, x=df2["Sex_name"], hue=df2["Survived_name"])
plt.show()
```

[81]



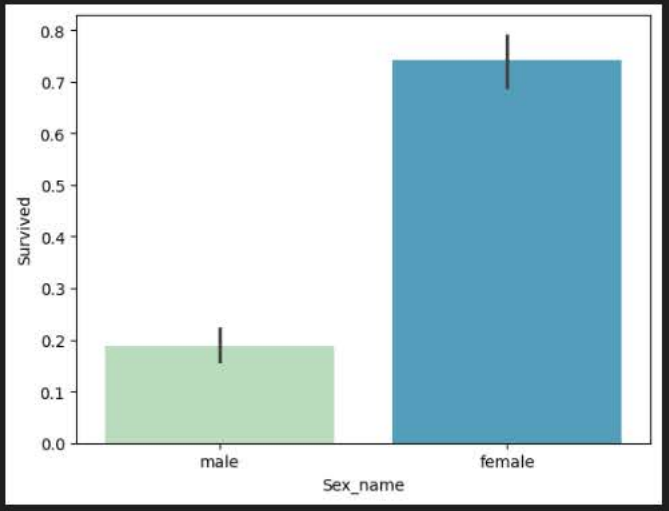
```
sns.barplot(data=df2, x=df2["Sex_name"], y=df2["Survived"], palette="GnBu")
plt.show()
```

[85]

```
C:\Users\SANDY\AppData\Local\Temp\ipykernel_13360\1234461620.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=df2, x=df2["Sex_name"], y=df2["Survived"], palette="GnBu")
```

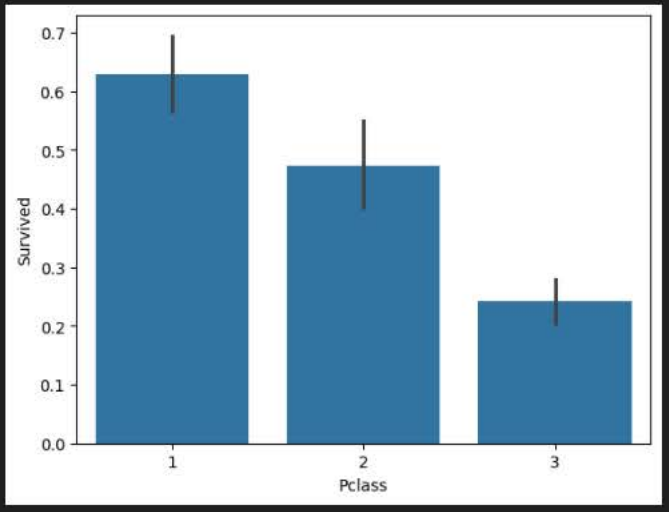


So, females have a better chance of survival as compared to males

Which Passenger Class Has Better Chance of Survival (First, Second, Or Third Class)?

```
df2.head(2)
sns.barplot(data=df2, x=df2["Pclass"], y=df2["Survived"])
plt.show()
```

[87]



So, First Class Passengers have the most chances of survival.

13. Feature Engineering - USED TO INCREASE EFFICENCY FOR ML ALGORITHMS

[91]
```
df2.columns
df2.head(2)
df2["Family_size"] = df2["SibSp"] + df2["Parch"]
df2.head(2)
```

...

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_Q	Embarked_S	Sex_name	Survived_name	Family_size
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	0	1	male	Died	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	2	38.0	1	0	PC 17599	71.2833	0	0	female	Survived	1

FARE PER PERSON

[ ]
```
df2["Fare_pre_person"] = df2["Fare"] / (df2["Family_size"] +1)
```