PROJECT NAME = TITANIC SURVIVAL PREDICTION

Importing Library ¶

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

Data

In [2]:
```
df=pd.read_csv("C:\\Users\\DISHA__COMPUTERS\\Desktop\\Internship\\1) titta
```

In [3]:
```
df
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298
...	...	...	...	...	...	...	...	...	...
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668

418 rows × 12 columns

EDA

In [4]:
```
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875

In [5]:
```
df.tail()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	10
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	2

In [6]:
```
df.describe()
```

Out[6]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Far
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.00000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.62718
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.90757
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.00000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.89580
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.45420
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.50000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.32920

In [7]:
```
type(df)
```

Out[7]:
```
pandas.core.frame.DataFrame
```

In [8]:
```
df.columns
```

Out[8]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibS
p',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [9]:
```
df.shape
```

Out[9]:
```
(418, 12)
```

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

Data Cleaning

```
# Checking Null Value and Duplicates
```

In [11]:
```
df.isna().sum()
```

Out[11]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [12]:
```
df['Age'] = df['Age'].fillna(df['Age'].mean())
df['Fare'] = df['Fare'].fillna(df['Fare'].mean())
```

```
Could not convert x to numeric of Cabin Column.
```

In [13]:
```
df.isna().sum()
```

Out[13]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          327
Embarked         0
dtype: int64
```

In [14]:
```
df.duplicated()
```

Out[14]:
```
0      False
1      False
2      False
3      False
4      False
       ...
413    False
414    False
415    False
416    False
417    False
Length: 418, dtype: bool
```

In [15]:
```
df.duplicated().sum()
```

Out[15]:
```
0
```

Convert object to numeric

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          418 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         418 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

In [17]:
```
Embarked = df['Embarked'].unique()
for Embarkeds in Embarked:
    print(Embarkeds)
```

```
Q
S
C
```

In [18]:
```
df['Embarked'] = df['Embarked'].map( {'Q': 0,'S':1,'C':2}).astype(int)
df['Sex'] = df['Sex'].map( {'female': 1,'male':0}).astype(int)
```

In [72]:
```
df['Age'] = df['Age'].astype(int)


df['Fare'] = df['Fare'].astype(int)
```

In [73]:
```
data = df.drop(['PassengerId','Name','Cabin','Ticket'], axis =1, inplace=T
```

In [74]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  418 non-null    int64
 1   Pclass    418 non-null    int64
 2   Sex       418 non-null    int32
 3   Age       418 non-null    int32
 4   SibSp     418 non-null    int64
 5   Parch     418 non-null    int64
 6   Fare      418 non-null    int32
 7   Embarked  418 non-null    int32
dtypes: int32(4), int64(4)
memory usage: 19.7 KB
```

In [75]:
```
df
```

Out[75]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	0	34	0	0	7	0
1	1	3	1	47	1	0	7	1
2	0	2	0	62	0	0	9	0
3	0	3	0	27	0	0	8	1
4	1	3	1	22	1	1	12	1
...	...	...	...	...	...	...	...	...
413	0	3	0	30	0	0	8	1
414	1	1	1	39	0	0	108	2
415	0	3	0	38	0	0	7	1
416	0	3	0	30	0	0	8	1
417	0	3	0	30	1	1	22	2

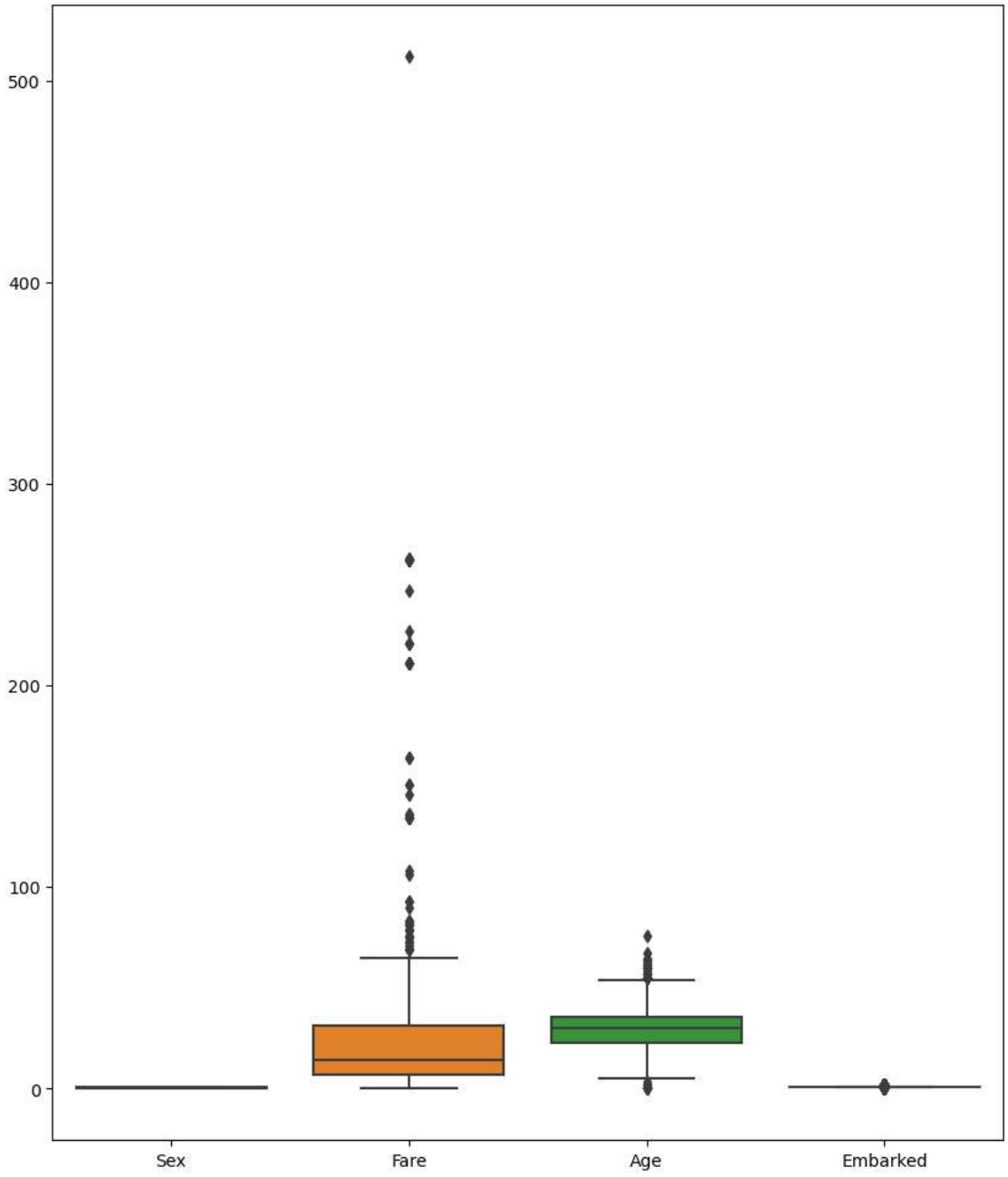
418 rows × 8 columns

Data Visualization

```
Outlier
```

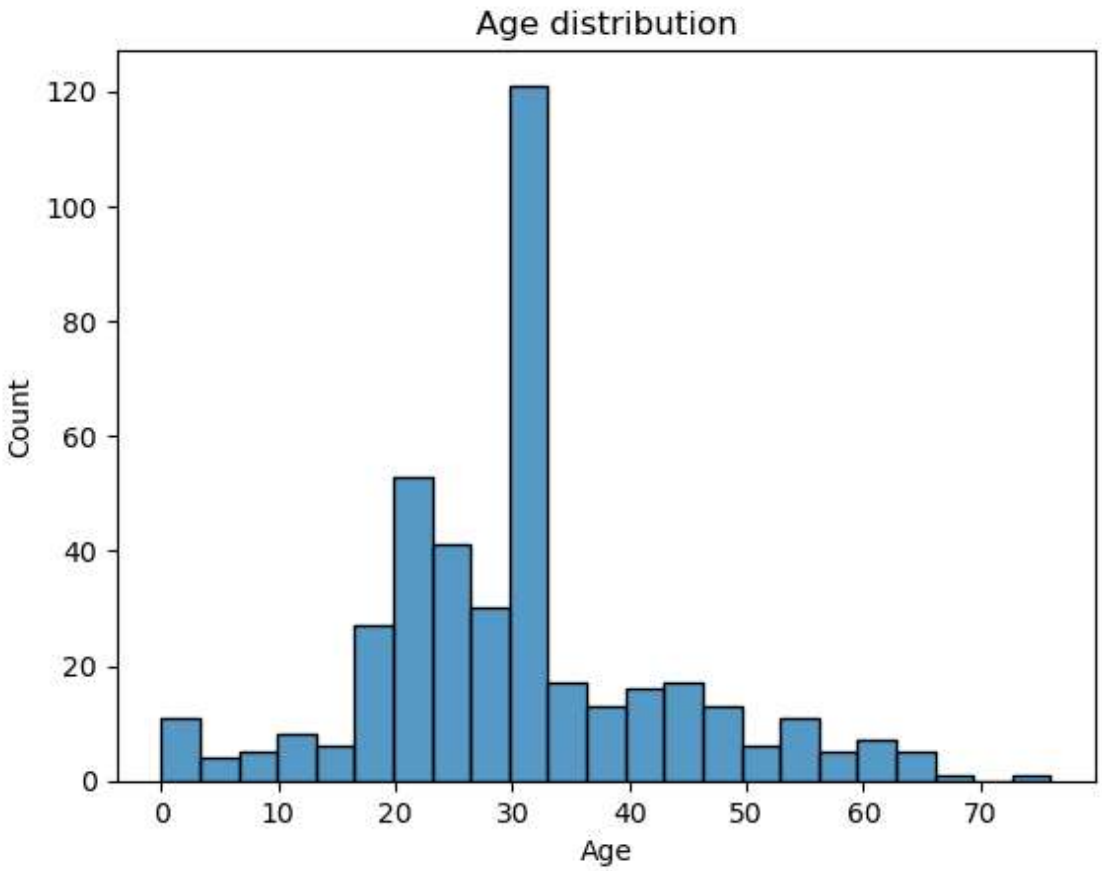
In [76]:
```
plt.figure(figsize=(10,12))
sns.boxplot(data=df[['Sex','Fare','Age','Embarked']])
```

Out[76]: <AxesSubplot:>



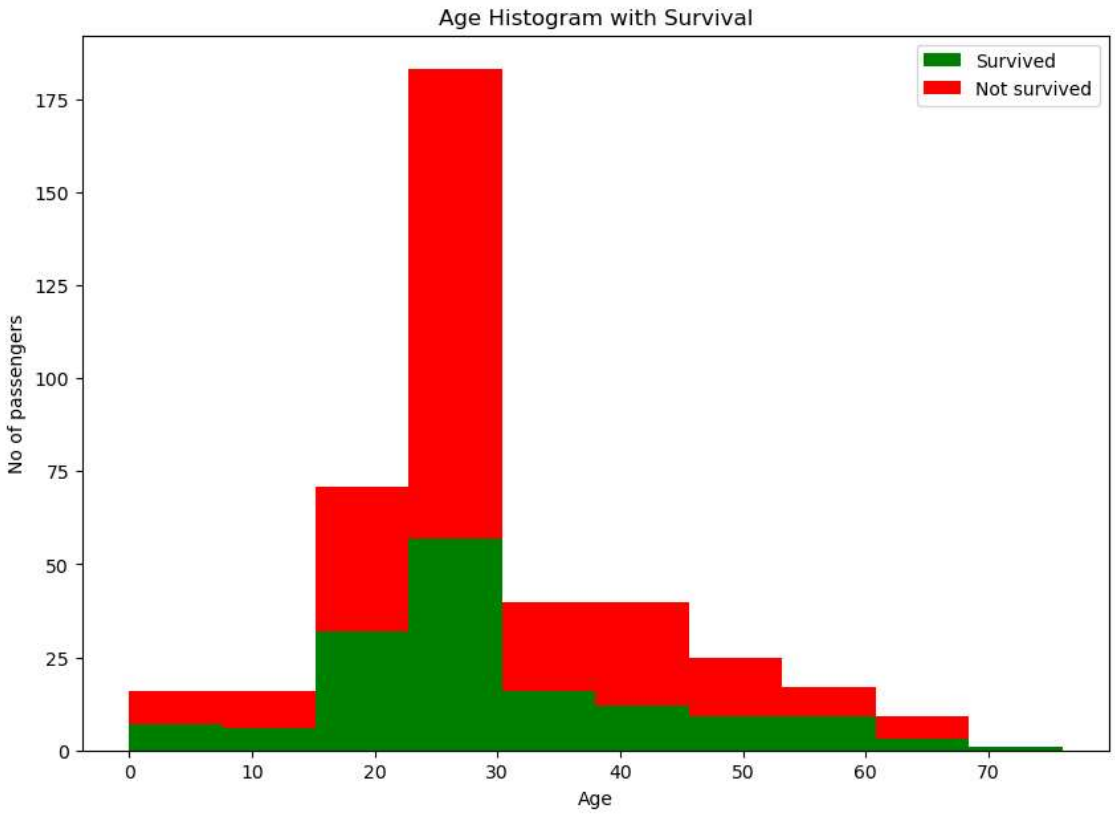
In [77]:
```
sns.histplot(df.Age)
plt.title('Age distribution')
```

Out[77]: Text(0.5, 1.0, 'Age distribution')



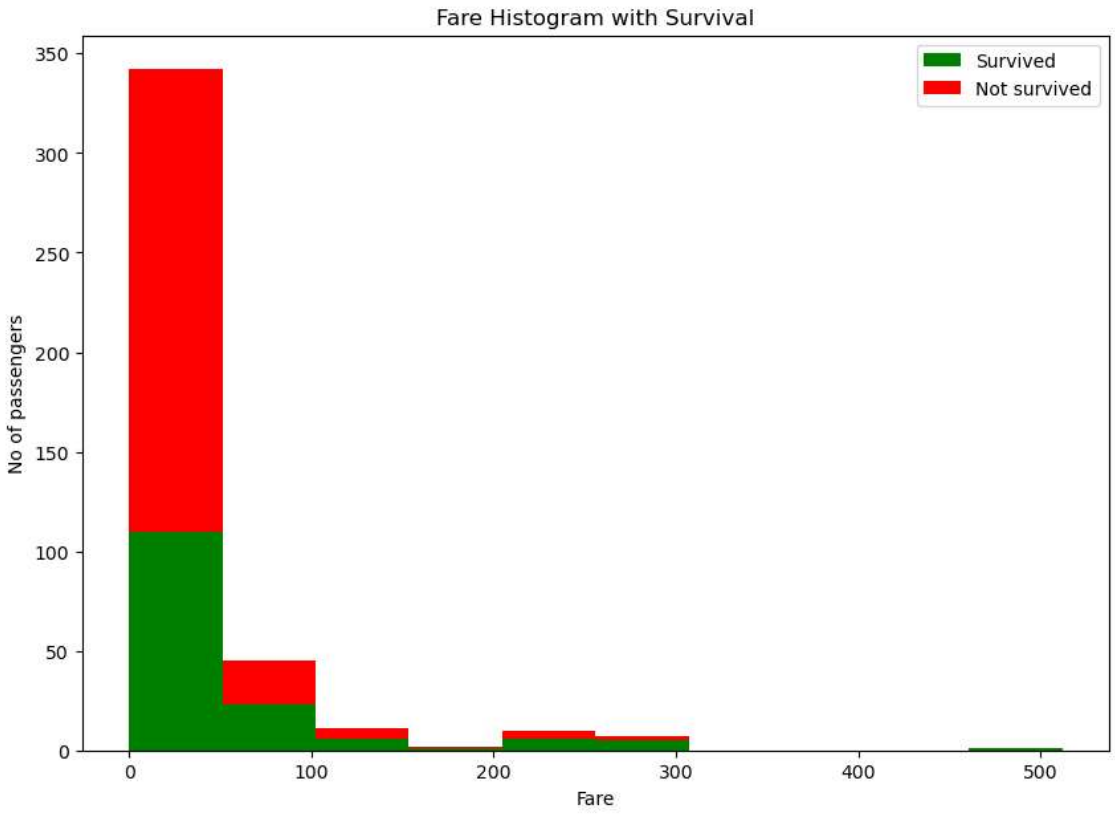
In [78]:
```
fig = plt.figure(figsize =(10, 7))
plt.hist(x = [df[df['Survived']==1]['Age'], df[df['Survived']==0]['Age']],
plt.title('Age Histogram with Survival')
plt.xlabel('Age')
plt.ylabel('No of passengers')
plt.legend()
```

Out[78]: <matplotlib.legend.Legend at 0x2ada26fb880>



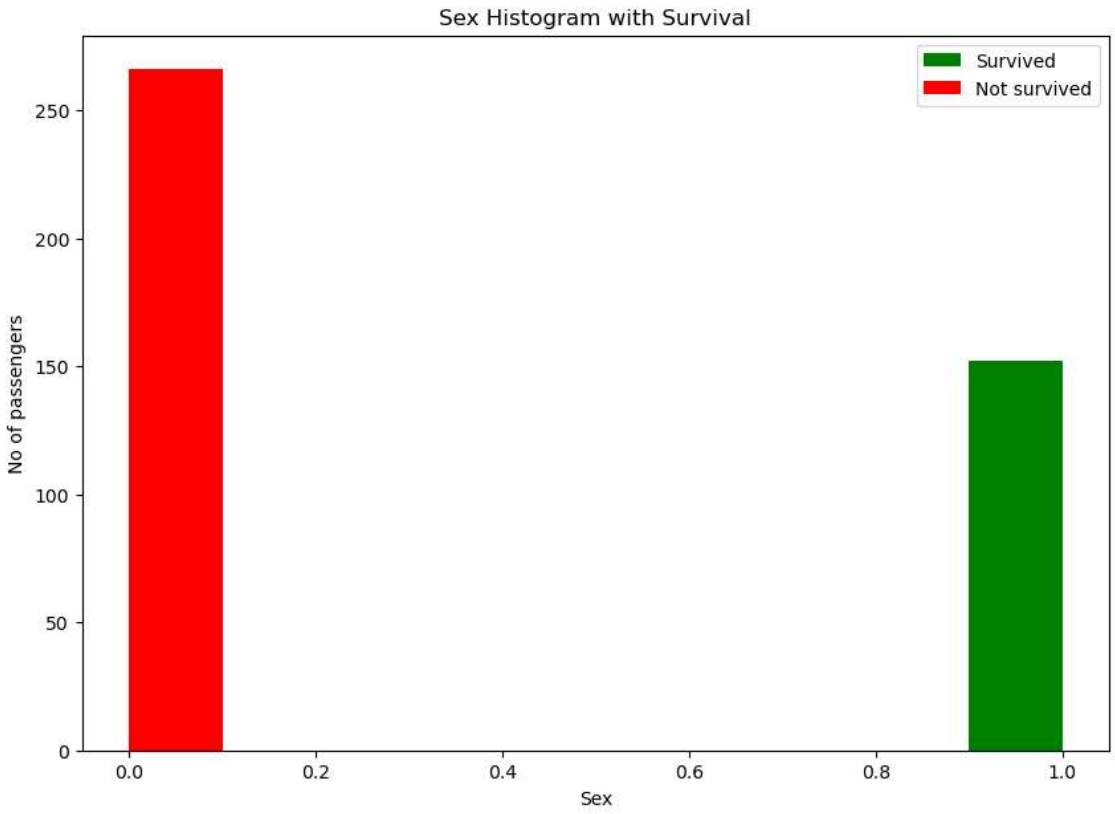
In [79]:
```
fig = plt.figure(figsize =(10, 7))
plt.hist(x = [df[df['Survived']==1]['Fare'], df[df['Survived']==0]['Fare']
plt.title('Fare Histogram with Survival')
plt.xlabel('Fare')
plt.ylabel('No of passengers')
plt.legend()
```

Out[79]: <matplotlib.legend.Legend at 0x2ada275b550>



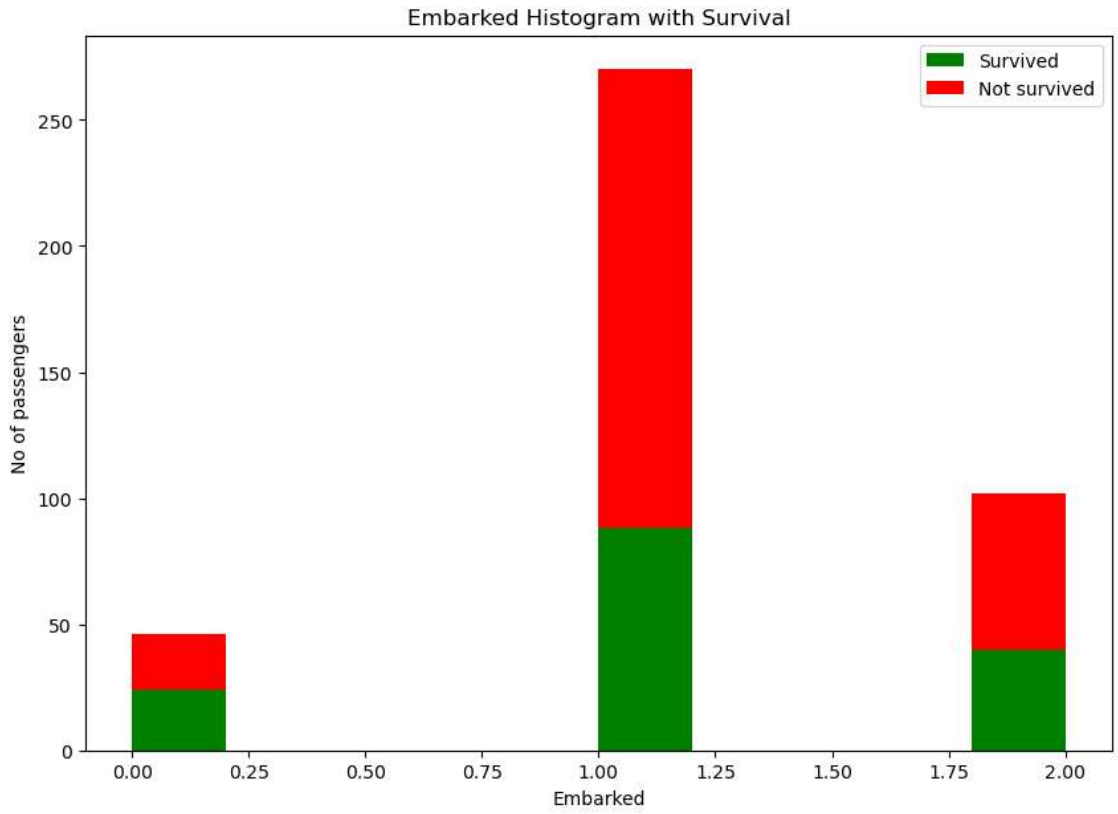
In [80]:
```
fig = plt.figure(figsize =(10, 7))
plt.hist(x = [df[df['Survived']==1]['Sex'], df[df['Survived']==0]['Sex']],
plt.title('Sex Histogram with Survival')
plt.xlabel('Sex')
plt.ylabel('No of passengers')
plt.legend()
```

Out[80]: <matplotlib.legend.Legend at 0x2ada2a76370>



In [81]:
```
fig = plt.figure(figsize =(10, 7))
plt.hist(x = [df[df['Survived']==1]['Embarked'], df[df['Survived']==0]['Em
plt.title('Embarked Histogram with Survival')
plt.xlabel('Embarked')
plt.ylabel('No of passengers')
plt.legend()
```

Out[81]: <matplotlib.legend.Legend at 0x2ada30b38b0>

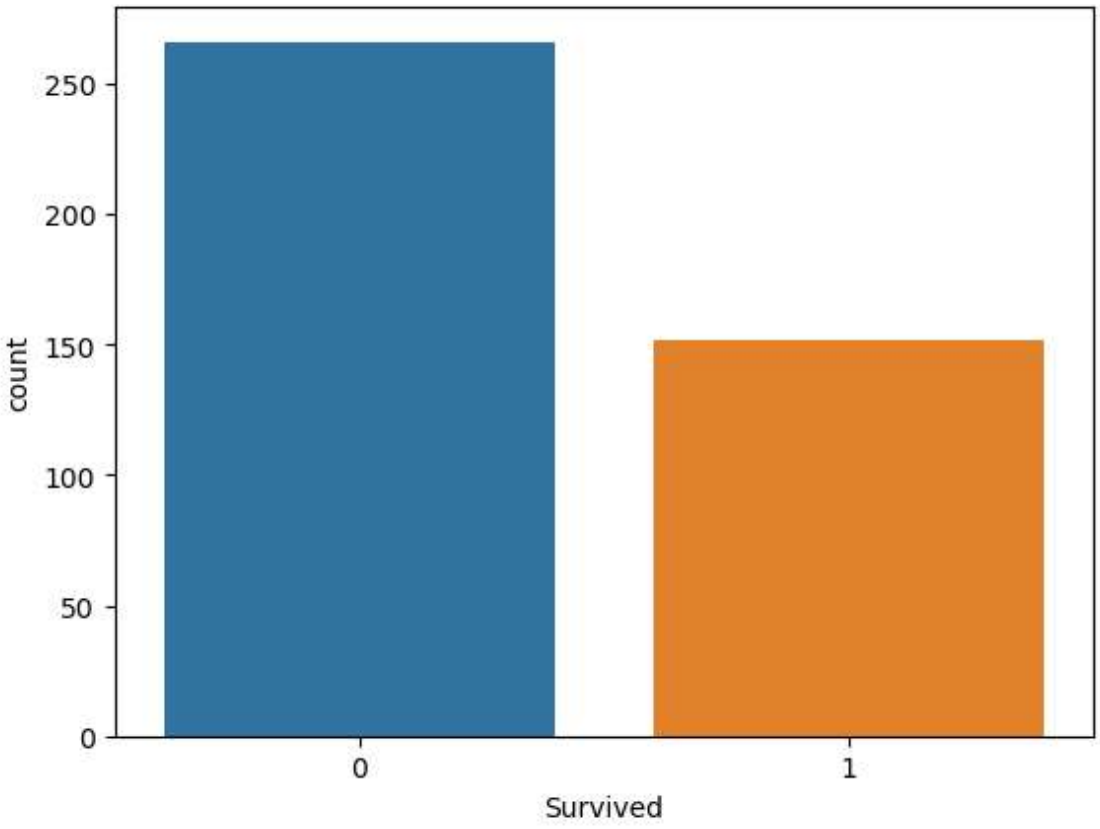


In [82]:
```
df.Survived.value_counts()
```

Out[82]:
```
0    266
1    152
Name: Survived, dtype: int64
```

In [83]:
```
sns.countplot('Survived',data=df)
```

Out[83]: <AxesSubplot:xlabel='Survived', ylabel='count'>



Splitting The data

In [84]:
```
x = df.drop(['Survived'], axis=1)
y = df.iloc[:,1]
```

In [85]:
```
x.shape
```

Out[85]: (418, 7)

In [86]:
```
from sklearn.model_selection import train_test_split
```

In [87]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2,
```

In [105]:
```

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

In [106]:
```
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

In [107]:
```
LR = LogisticRegression(solver='liblinear', max_iter=200)
LR.fit(x_train, y_train)
y_pred = LR.predict(x_test)
LR1 = accuracy_score(y_pred,y_test)
print('Logistic regression accuracy: {:.2f}%'.format(LR1*100))
```

```
Logistic regression accuracy: 92.86%
```

In [108]:
```
svc_model=SVC()
svc_model.fit(x_train,y_train)
pred=svc_model.predict(x_test)
```

In [109]:
```
SVC = accuracy_score(y_pred,y_test)
print('SVC accuracy: {:.2f}%'.format(LR1*100))
```

```
SVC accuracy: 92.86%
```

In [111]:
```
RF1=RandomForestClassifier()
RF1.fit(x_train,y_train)
pred=RF1.predict(x_test)
```

In [113]:
```
Rf1= accuracy_score(y_pred,y_test)
print('RandomForestClassifier accuracy: {:.2f}%'.format(LR1*100))
```

```
RandomForestClassifier accuracy: 92.86%
```

In [ ]:

In [ ]: