```
Assignment-3 (Anish Singh) 21BCE3775
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

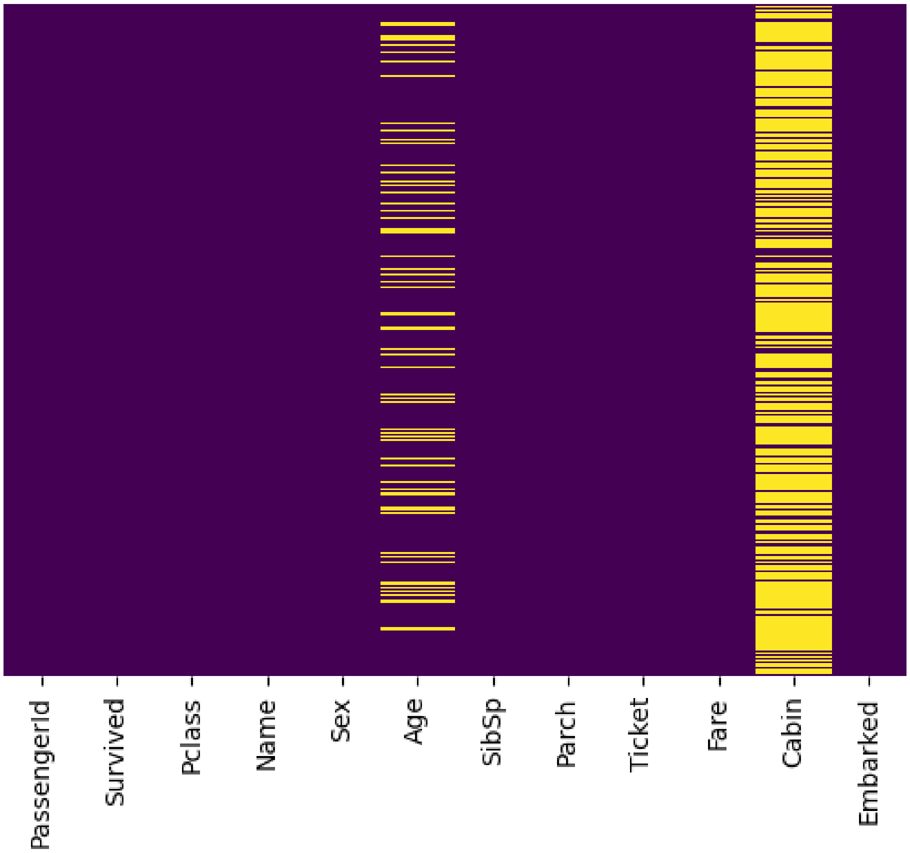
In [4]:
```
data = pd.read_csv('Titanic-Dataset.csv')
data.head()
```

Out[4]:

	PassengerId	Survived	Pclass	N me	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Bra nd, Mr. C en H rris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumi gs, 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

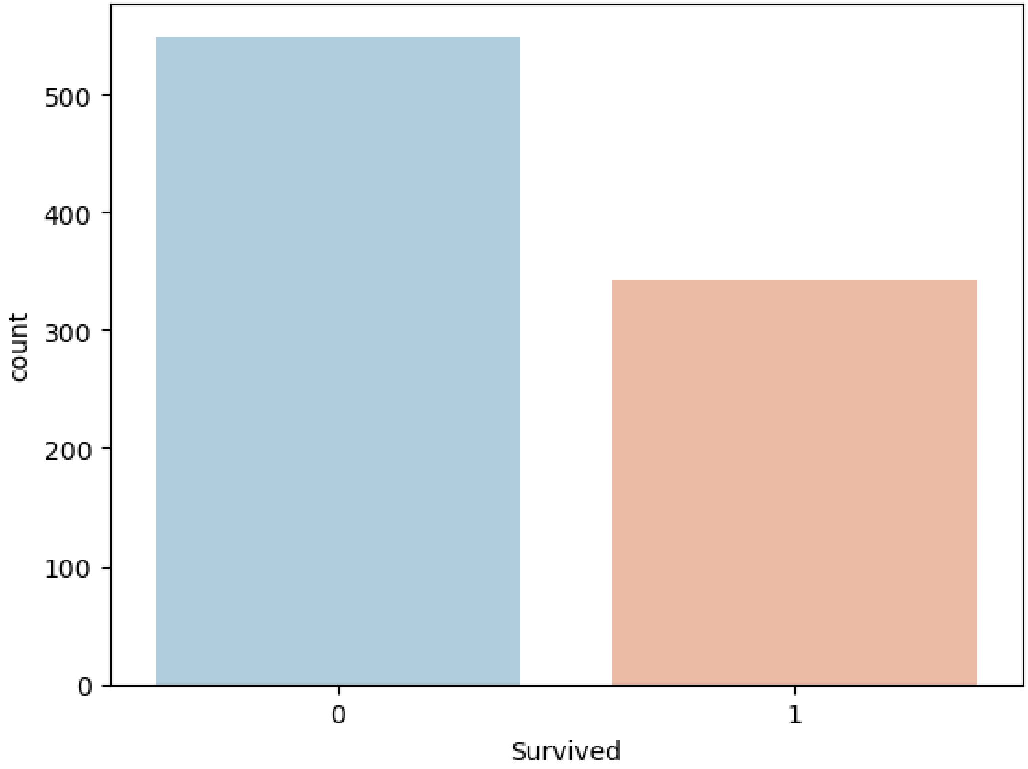
In [5]:
```
sns.heatmap(data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[5]:
```
<Axes: >
```



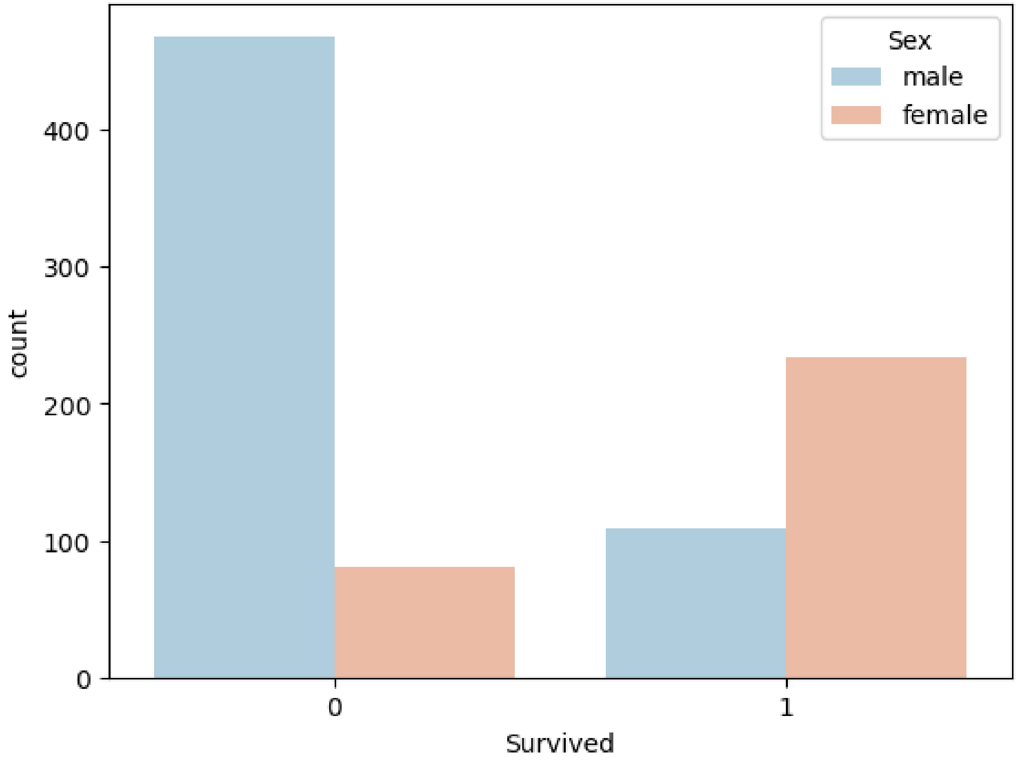
```
In [6]: sns.countplot(x='Survived',data=data,palette='RdBu_r')
```

```
Out[6]: <Axes: xlabel='Survived', ylabel='count'>
```



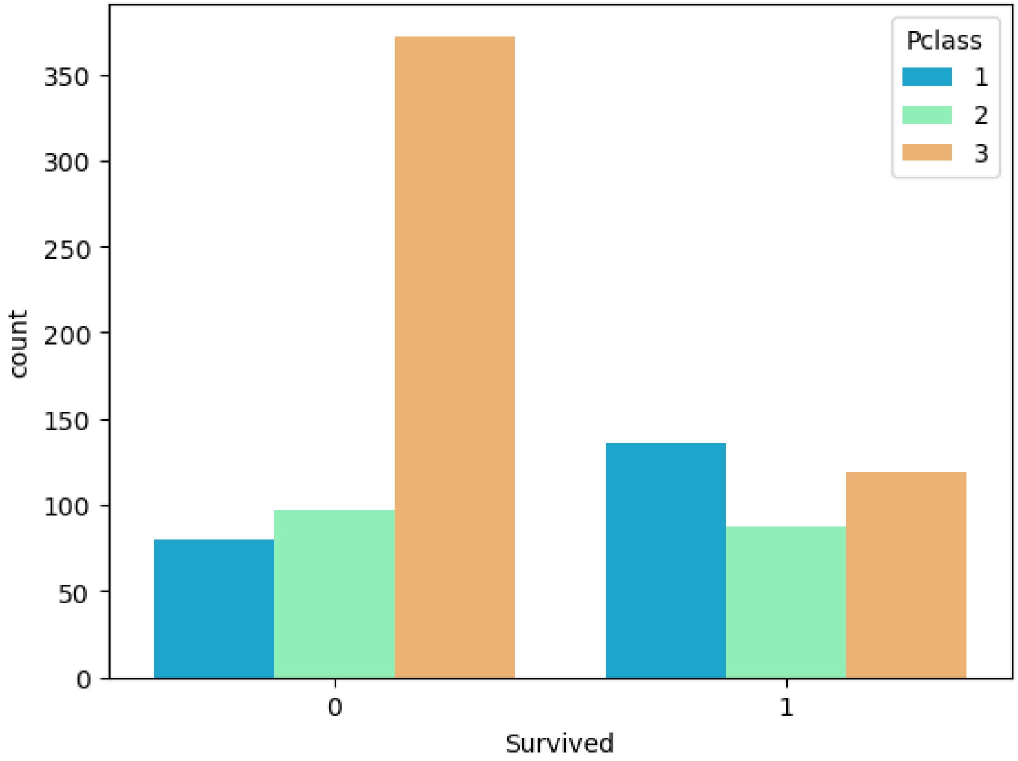
```
In [7]: sns.countplot(x='Survived',hue='Sex',data=data,palette='RdBu_r')
```

```
Out[7]: <Axes: xlabel='Survived', ylabel='count'>
```



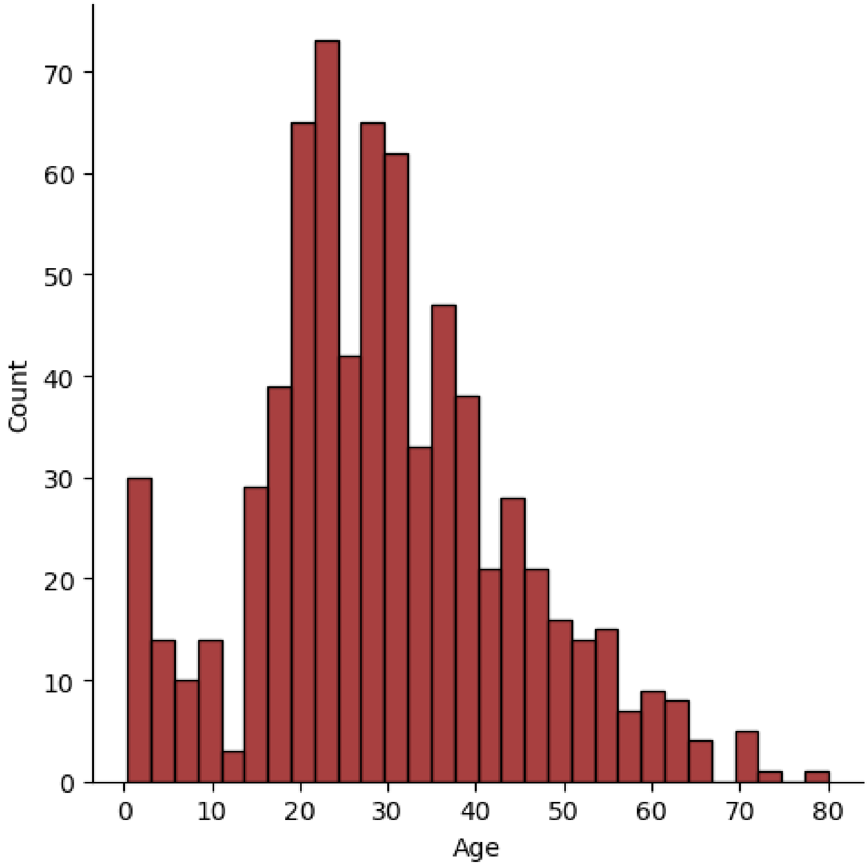
```
In [8]: sns.countplot(x='Survived',hue='Pclass',data=data,palette='rainbow')
```

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



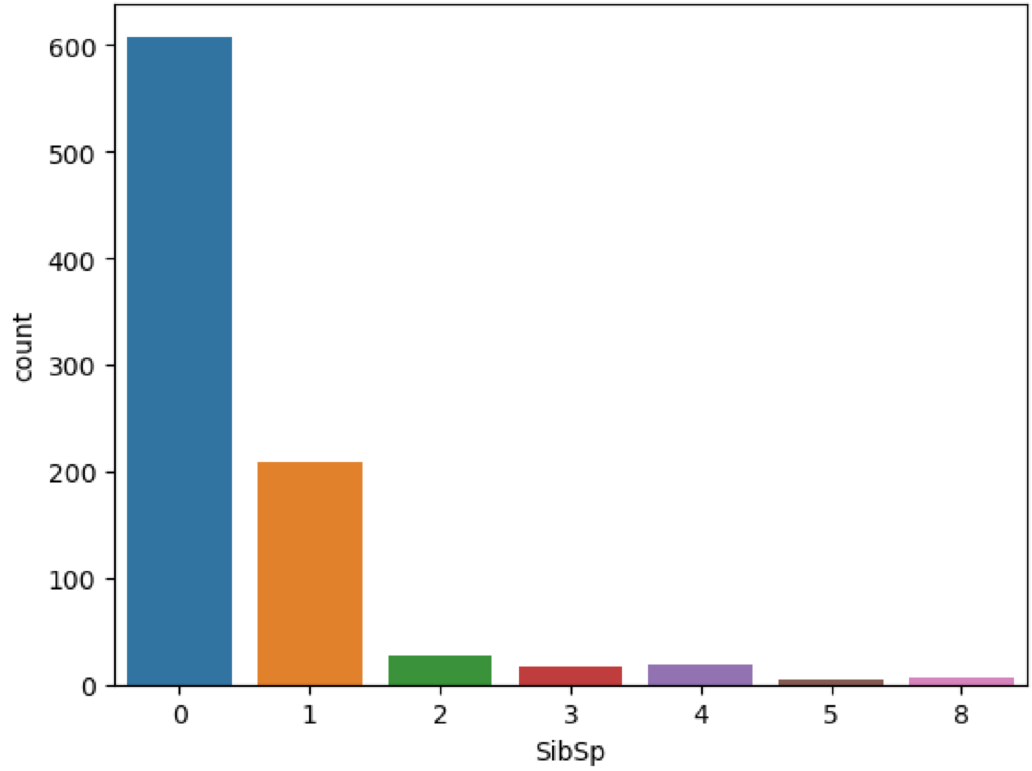
In [11]:
```
sns.displot(data['Age'].dropna(),kde=False,color='darkred',bins=30)
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x209004ef790>
```



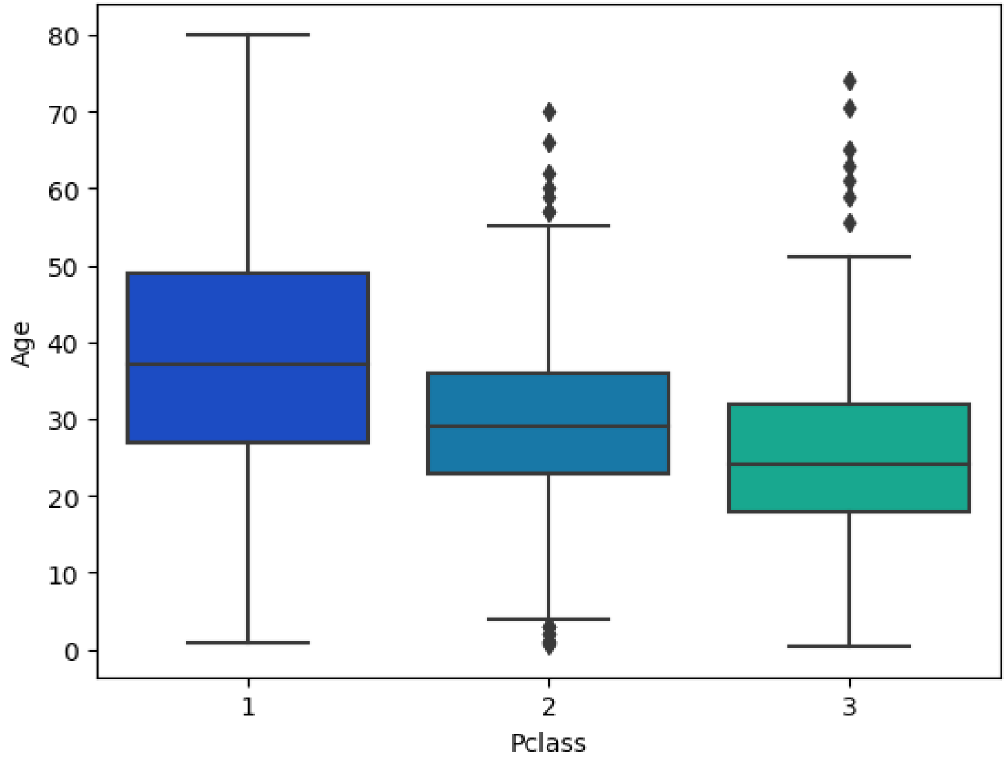
In [12]:
```
sns.countplot(x='SibSp',data=data)
```

Out[12]:
```
<Axes: xlabel='SibSp', ylabel='count'>
```



In [13]:
```
sns.boxplot(x='Pclass',y='Age',data=data,palette='winter')
```

Out[13]:
```
<Axes: xlabel='Pclass', ylabel='Age'>
```

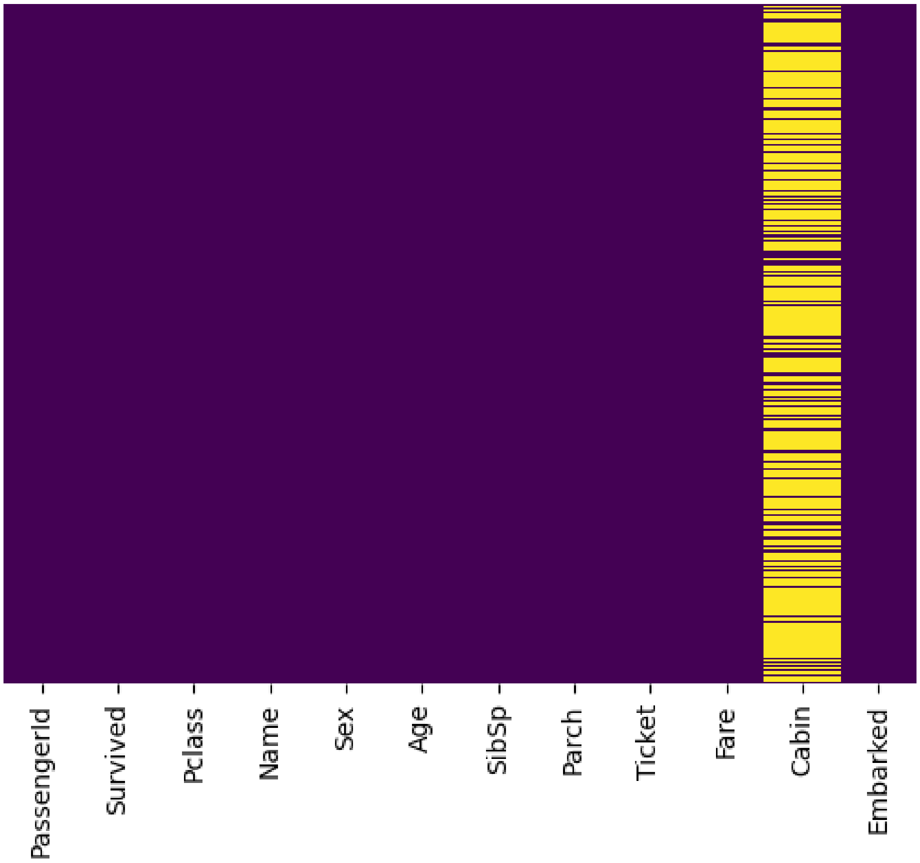


In [20]:
```
def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]

    if pd.isnull(Age):

        if Pclass == 1:
            return 37

        elif Pclass == 2:
            return 29

        else:
            return 24

    else:
        return Age
```

In [21]:
```
data['Age'] = data[['Age','Pclass']].apply(impute_age,axis=1)
```

In [22]:
```
sns.heatmap(data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[22]:
```
<Axes: >
```



In [23]:
```
data.drop('Cabin',axis=1,inplace=True)
```

In [24]:
```
data.head()
```

Out[24]:

	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 [25]:
```
data.dropna(inplace=True)
```

In [26]:
```
data.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 [27]:
```
sex = pd.get_dummies(data['Sex'],drop_first=True)
```

In [28]:
```
embark = pd.get_dummies(data['Embarked'],drop_first=True)
```

In [29]:
```
data.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace=True)
```

In [30]:
```
data = pd.concat([data,sex,embark],axis=1)
```

In [31]:
```
data.head()
```

Out[31]:

	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

In [87]:
```
from sklearn.model_selection import train_test_split
```

In [88]:
```
X = train.drop('Survived',axis=1)
y = train['Survived']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101
```

In [89]:
```
from sklearn.linear_model import LinearRegression
```

In [107]:
```
lm = LinearRegression()
```

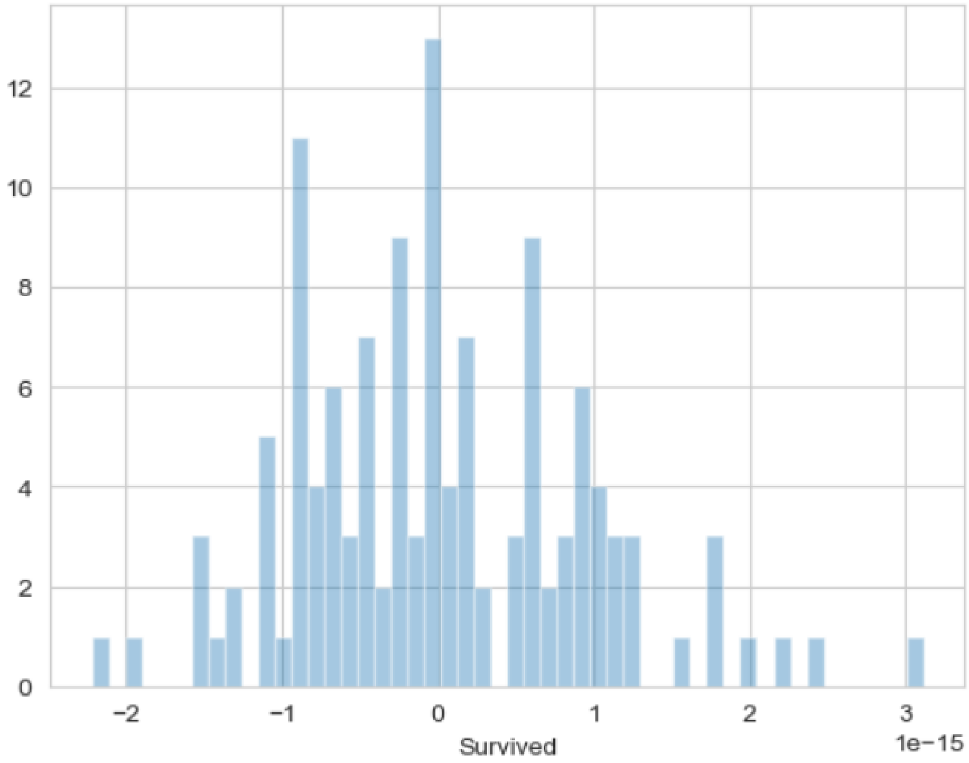
In [108]:
```
Lm.fit(X_train,y_train)
```

Out[108:

```
▾ LinearRegression
LinearRegression()
```

In [ ]:
```
predictions=lm.predict(X_test)
```

In [110]:
```
sns.displot((y_test-predictions),bins=50,kde=False);
```



In [111]:
```
from sklearn import metrics
```

In [112]:
```
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 7.128336721601204e-16
MSE: 8.392407869408556e-31
RMSE: 9.16100860681211e-16
```

In [ ]: