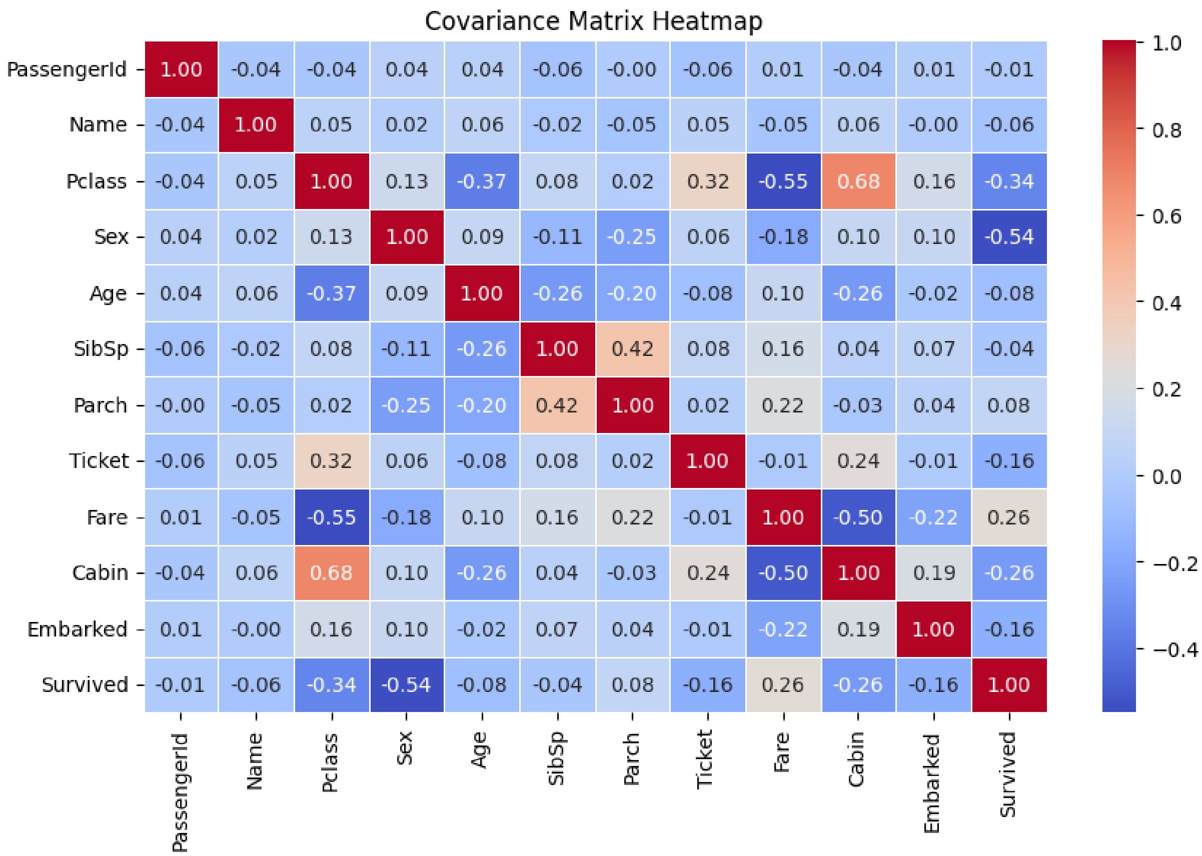
Intitial Covarience Matrix

Justification: From the picture we can see how the relationship with our target column (Survived) with all the features.

In [261…

```
cov_mat(new_df)
```



Handeling missing values

Justification: From this picture we can say, We have some NULL values in Age, Cabin, Embarked.

In [262…

```
df.isna().sum()
```

Out[262…

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

In [ ]:

```

```

Justification:

```
Age-> We decided to go with mean().

Embarked-> We decide to go with mode().

Cabin-> We fill all the Null values with a new word
"Unknown".
```

In [263…

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

In [265…

```
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

In [266…

```
df["Cabin"].fillna("Unknown",inplace=True)
```

We convert the ages into groups since the survival ratio lies between certain ages of people. So we'll find a clear understanding.

0->Newborn (0-4)

1->Kid (5-10)

2->Teenager (11-17)

3->Young Adults (18-28)

4->Middle ages (28-39)

5->Senior Citizen (40-rest)

In [271…

```
def categorize_age(age):
    if age<5:
        return 0.0
    elif age>=5 and age<11:
        return 1.0
    elif age>=11 and age<18:
```

```
        return 2.0
    elif age>=18 and age<28:
        return 3.0
    elif age>=28 and age<39:
        return 4.0
    else:
        return 5.0
```

We narrow the number of Siblings/Spouses into three groups since the survival ratio lies between two to three different kinds of people. So we'll find a clear understanding.

0->Solo (0)

1->Couple/Small Group (1-3)

2->Group (4-rest)

We also narrowed the number of Parents/Children into three groups since the survival ratio lies between two to three different kinds of people. So we'll find a clear understanding.

0->Single (0)

1->Nuclear Family (1-3)

2->Joint family (4-rest)

In [ ]:
```
def family_type(type):
    if type<=0:
        return 0
    elif type>=1 and type<4:
        return 1
    else:
        return 2
```

We convert all the fares into 5 groups. So we'll find a better understanding.

0->Lower Economy Class (0-10.5)

1->Economoy Class (10.6-14.5)

2->Higher Economy Class (14.6-16)

3->Middle Class (16.1-55)

4->Business Class (56-rest)

In [309…
```
def fare_class(fare):
    if fare<=10.5:
        return 0
    elif 10.5<fare<=14.5:
        return 1
```

```
    elif 14.5<fare<16:
        return 2
    elif 16<fare<=55:
        return 3
    else:
        return 4
```

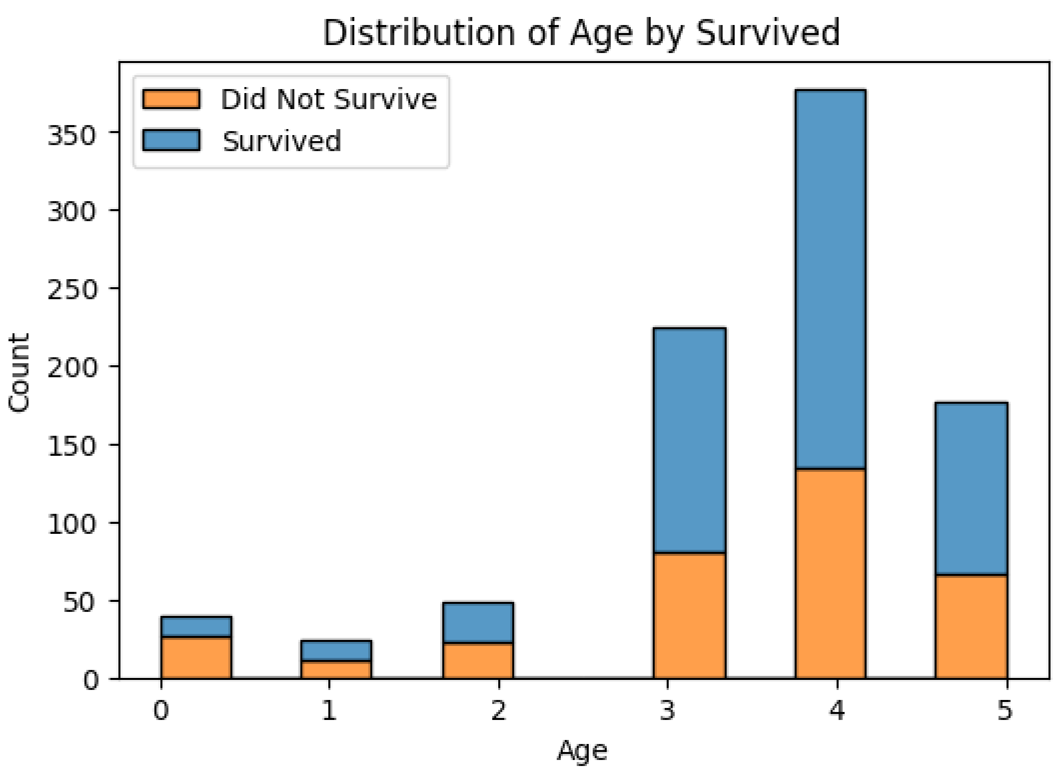
We narrow down all the Cabin values with two categories. Since a huge amount of NULL data is contributing in the column which is replace with the "Unknown" word.

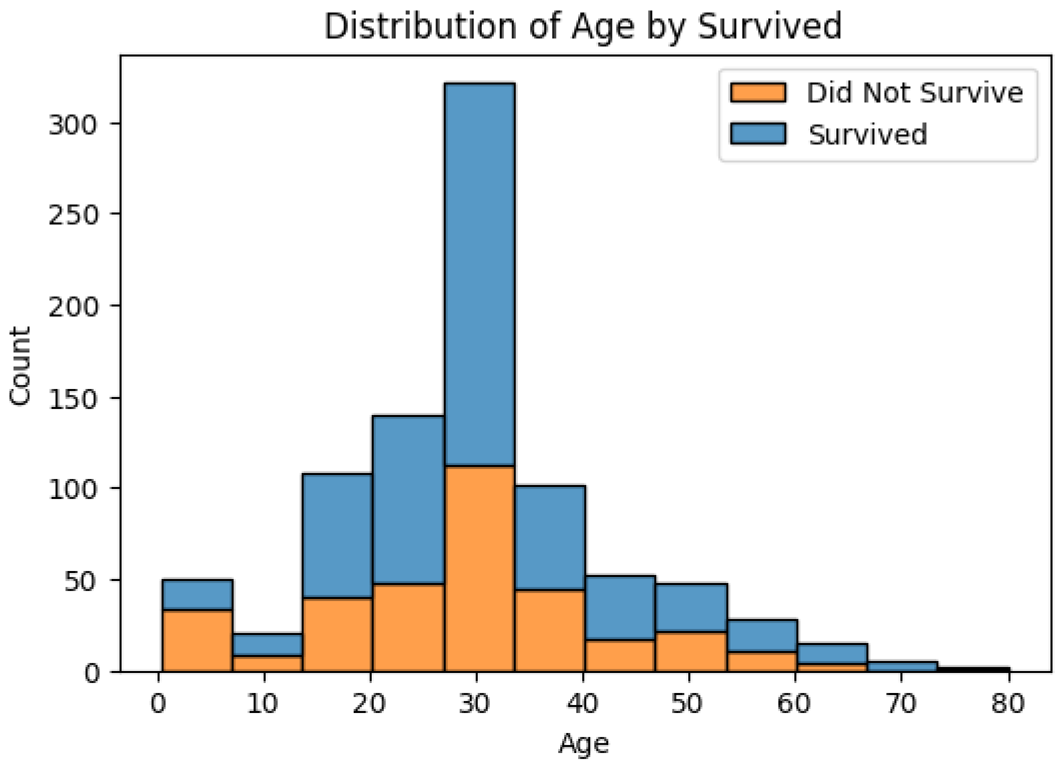
0->Unknown

1->Known

In [ ]:
```
def cabin(value):
    if value=="Unknown":
        return 0
    else:
        return 1
```

In [313…
```
print(visualization(feature_df,"Age","Survived"),visualization(df,"Age","Sur
```



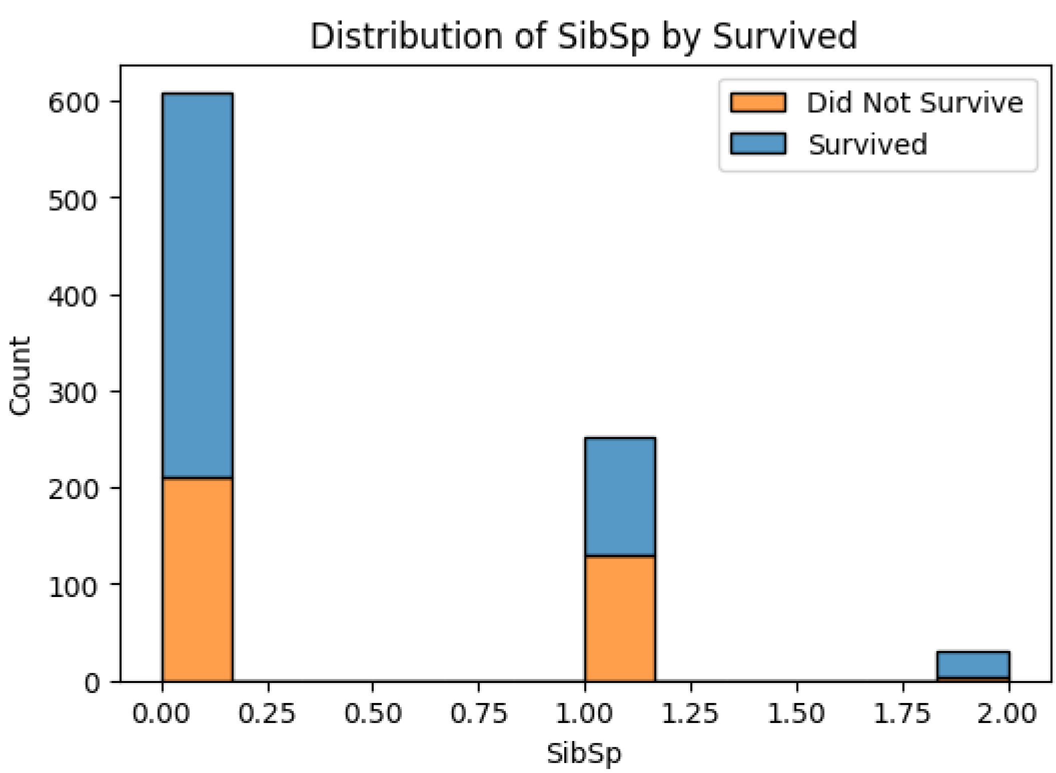


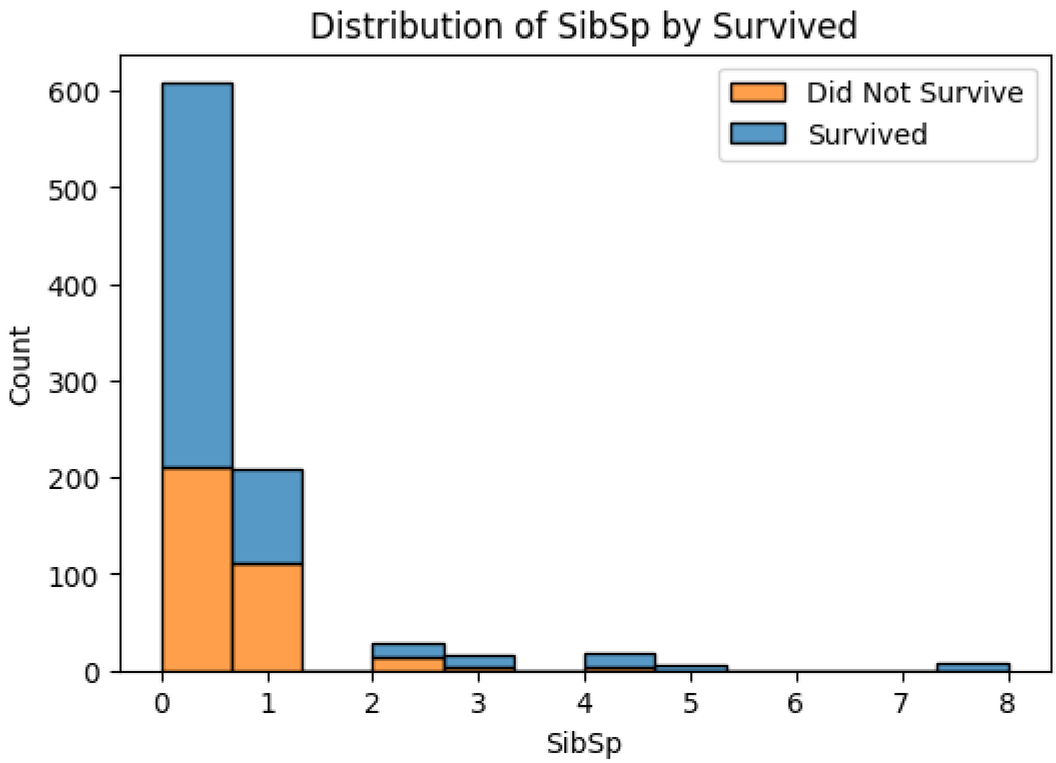

```
None None
```

Justification: First Picture is after feature engginerring and the last picture is without it. From that first picture we can say most of the people survive who are in Middle ages (28-39).

In [314…

```
print(visualization(feature_df,"SibSp","Survived"),visualization(df,"SibSp",
```



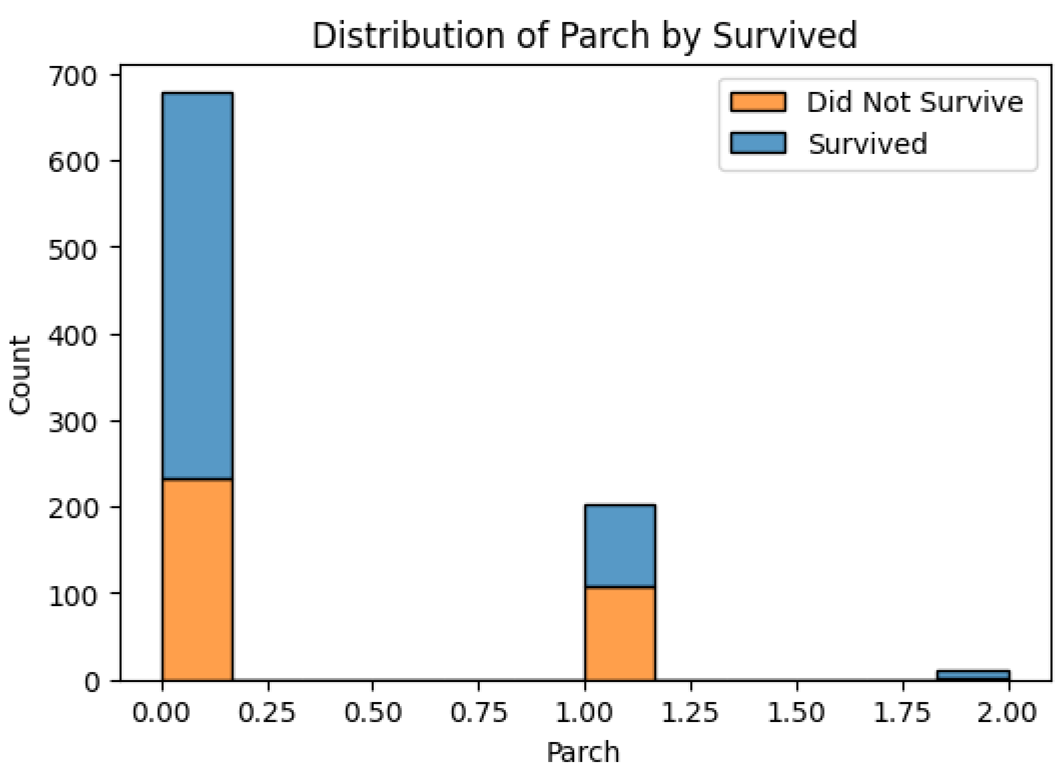


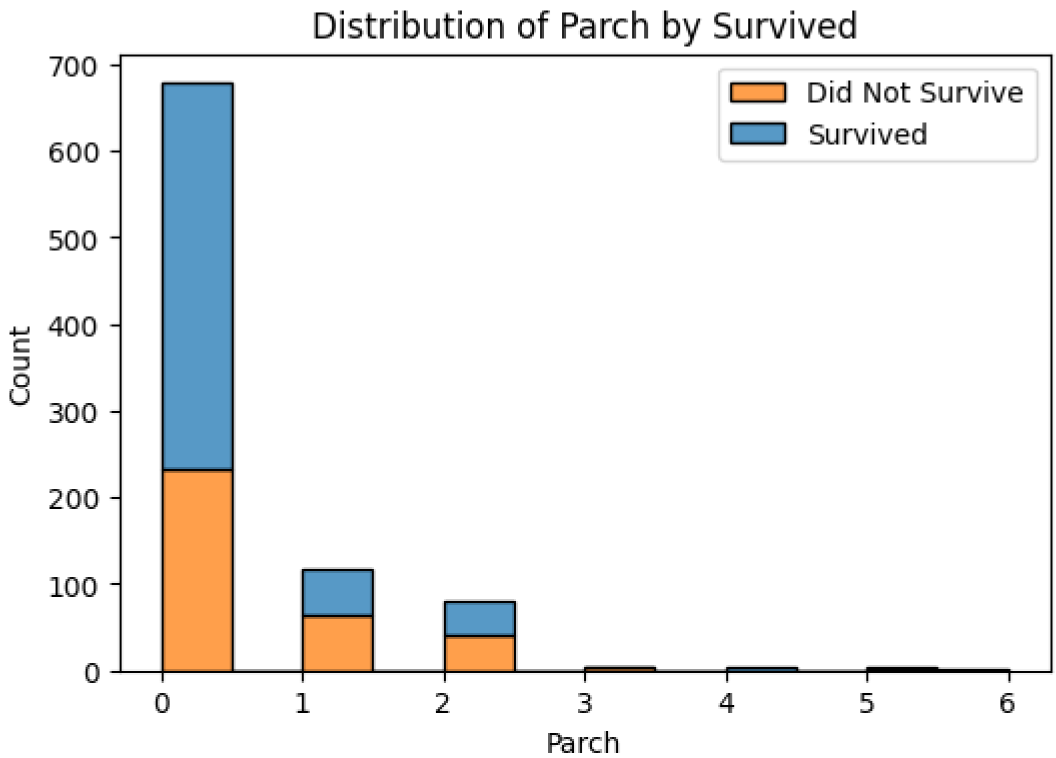

```
None None
```

Justification: First Picture is after feature enggineerring and the last picture is without it. From that first picture we can say most of the people survive who travel SOLO (0).

In [315…

```
print(visualization(feature_df,"Parch","Survived"),visualization(df,"Parch",
```

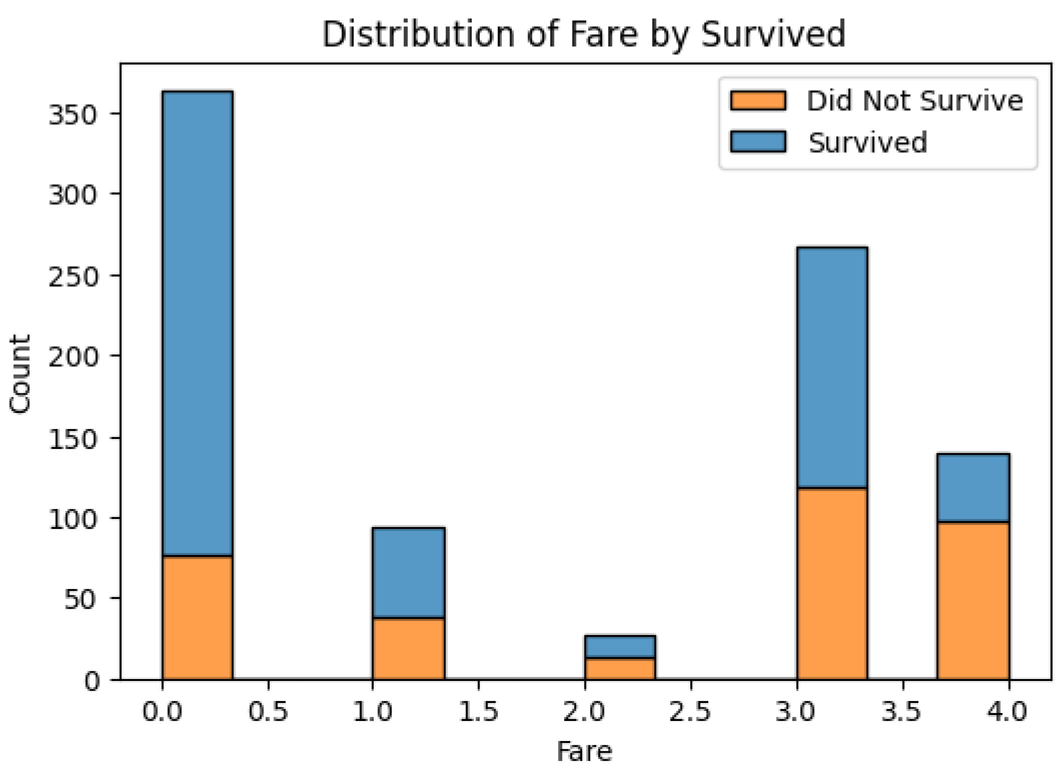


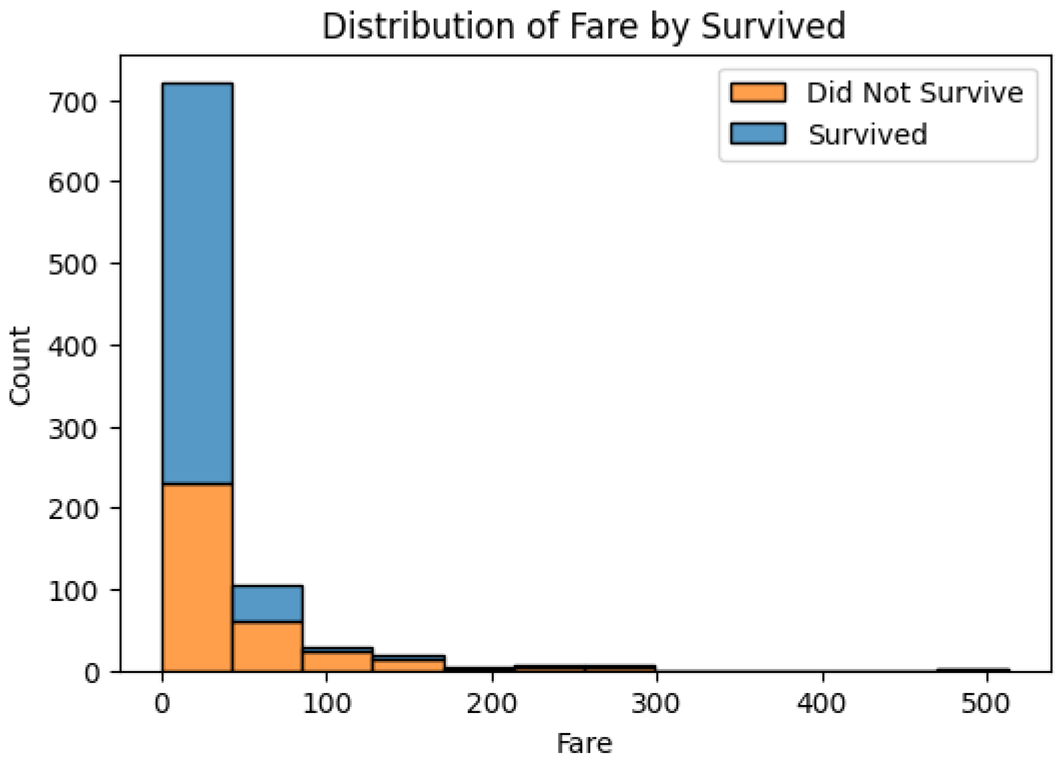



```
None None
```

Justification: The first picture is after the feature engineering and the last picture is without it. From that first picture, we can say most of people survive they are traveling Single (0).

```
In [316…  print(visualization(feature_df,"Fare","Survived"),visualization(df,"Fare","S
```



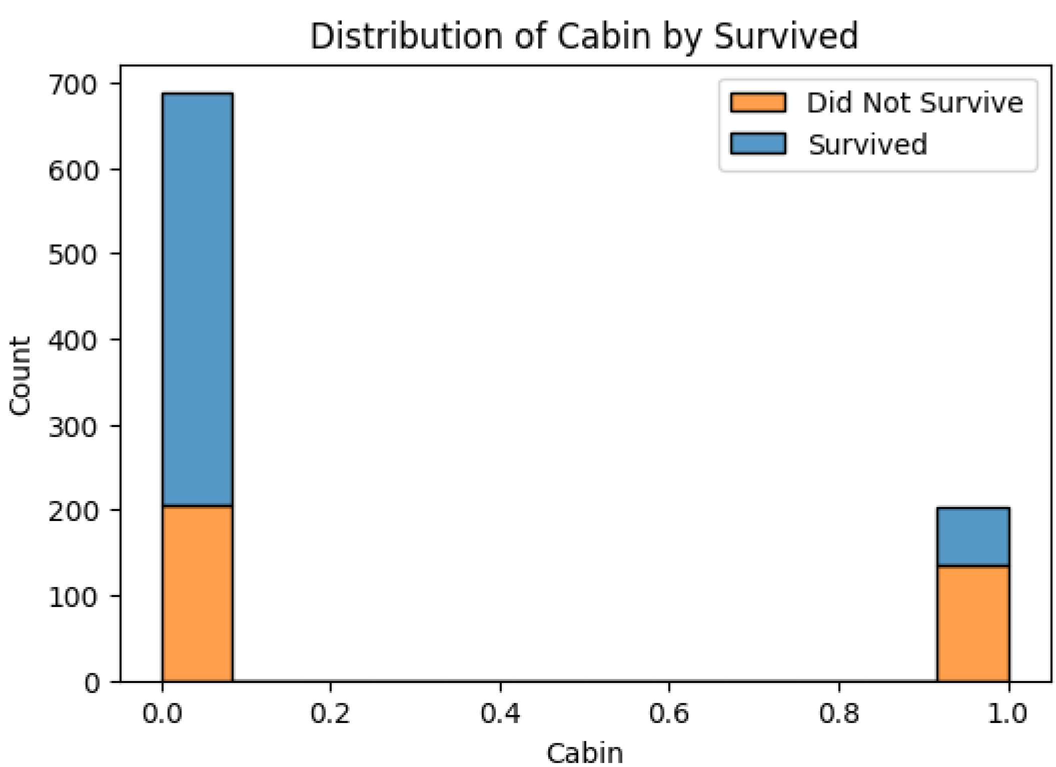


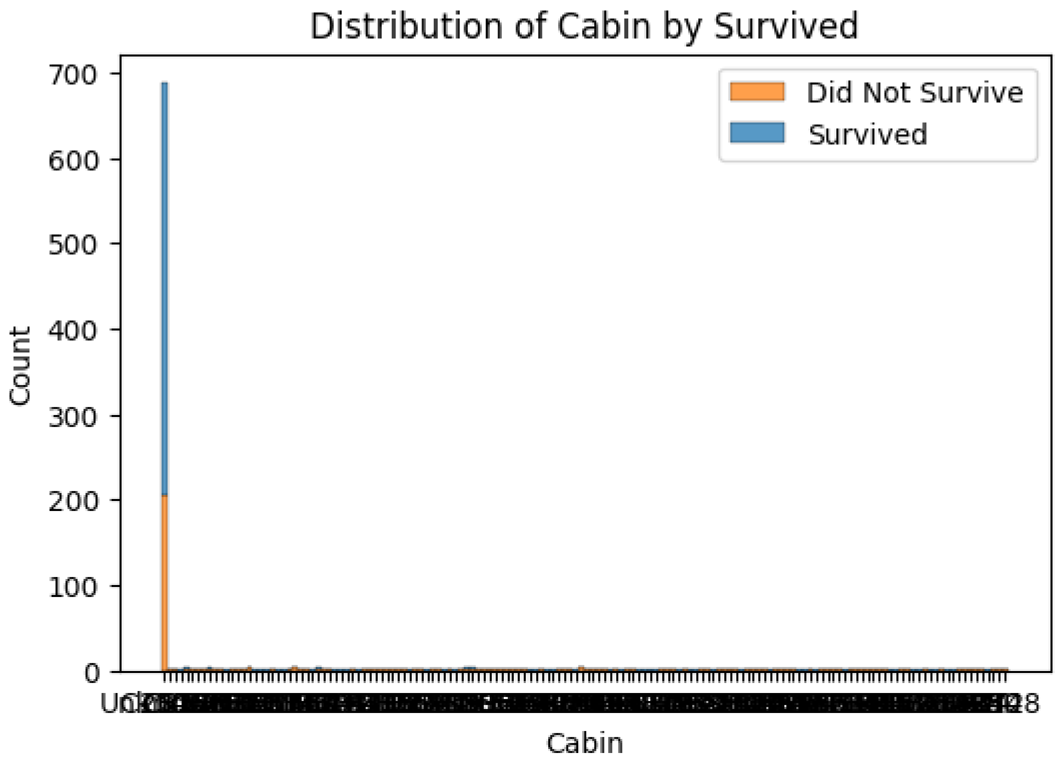

```
None None
```

Justification: The first picture is after the feature engineering and the last picture is without it. From that first picture, we can say most people survive who actually traveled with a lower economy class (0).

In [317…

```
print(visualization(feature_df,"Cabin","Survived"),visualization(df,"Cabin",
```



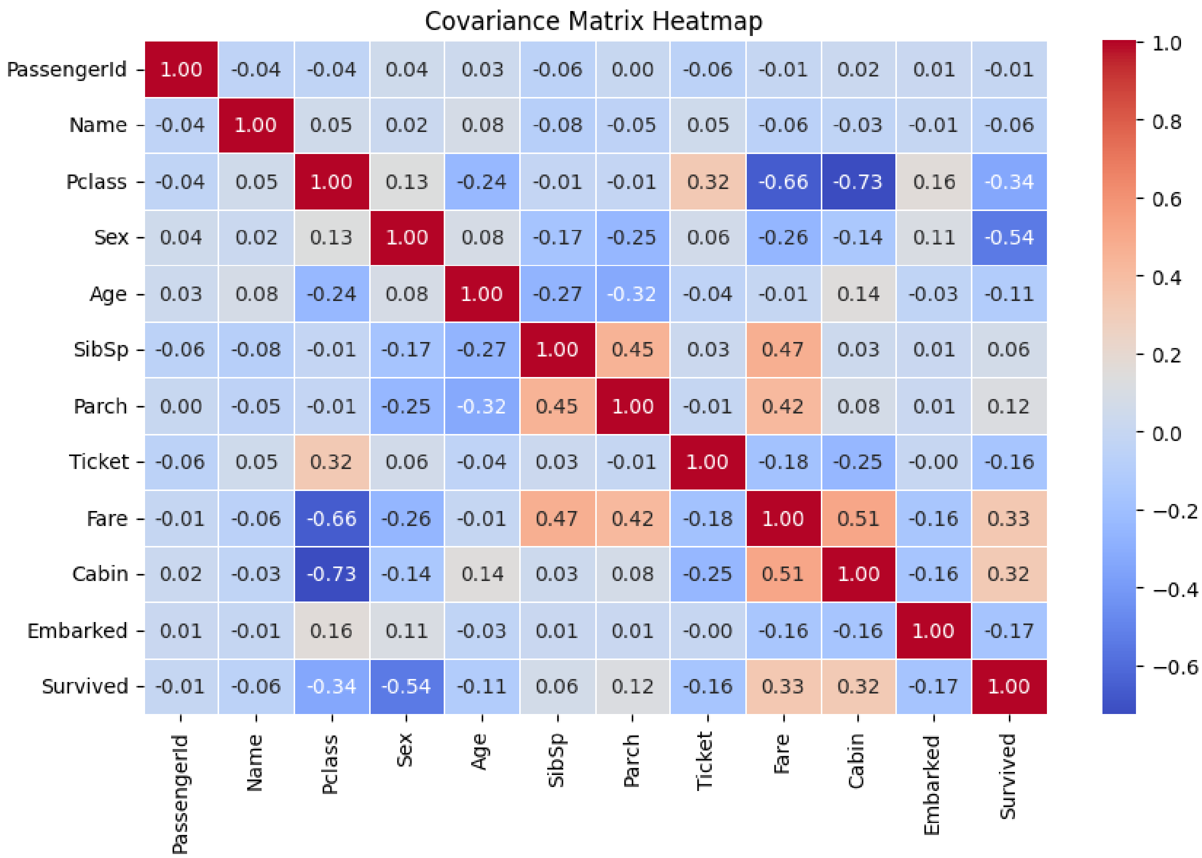



```
None None
```

Justification: The first picture is after the feature engineering and the last picture is without it. From that first picture, we can say most of the people who survive are "Unknown" (0) cabin have.

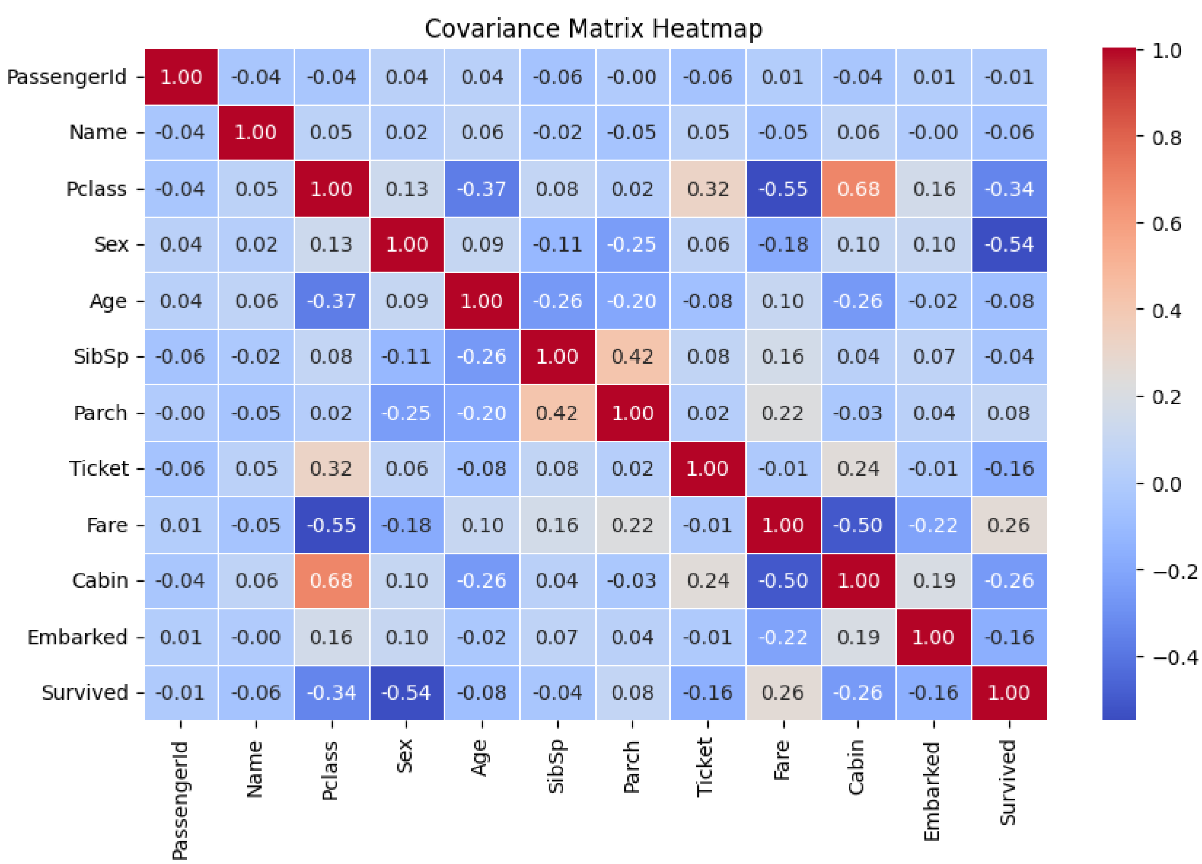
In [319…

```
cov_mat(feature_df)
```



In [291…

```
cov_mat(new_df)
```



Justification:

We can clearly see how covariance of various Features have changed drastically with our Target variable. Some observations are as follow,

Age(-0.08 to -0.11)

SibSp(-0.04 to 0.06)

Parch(-0.08 to 0.12)

Fare(0.26 to 0.33)

Cabin(-0.26 to 0.32)

In [ ]:
```

```