```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
```

```
In [2]: df = pd.read_csv(r"D:\New folder (2)\titanic_train.csv")
```

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

Out[3]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

```
In [4]: 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 [13]: df.shape

Out[13]: (891, 12)
```

```
In [14]: df.isna().sum()

Out[14]: PassengerId         0
Survived            0
Pclass              0
Name                0
Sex                 0
Age               177
SibSp               0
Parch               0
Ticket              0
Fare                0
Embarked            2
passenger_title     0
dtype: int64
```

```
In [15]: round(df.isna().sum()/len(df)*100)

Out[15]: PassengerId        0.0
Survived           0.0
Pclass             0.0
Name               0.0
Sex                0.0
Age               20.0
SibSp              0.0
Parch              0.0
Ticket             0.0
Fare               0.0
Embarked           0.0
passenger_title    0.0
dtype: float64
```

Missing Values

```
In [4]: df.drop('Cabin',axis=1,inplace=True)
```

```
In [5]: df.Name.head(50)
df['passenger_title'] = df.Name.apply(lambda x: x.split(',')[1].split('.')[0].strip())
```

```
In [7]: df['Age']=df['Age'].fillna(df.Age.median())
df['Embarked']=df['Embarked'].fillna(df.Embarked.mode()[0])
df.isna().sum()

Out[7]: PassengerId        0
Survived           0
Pclass             0
Name               0
Sex                0
Age                0
SibSp              0
Parch              0
Ticket             0
Fare               0
Embarked           0
passenger_title    0
dtype: int64
```

```
In [8]: df.drop('Name',axis=1,inplace=True)
df.head()
```

Out[8]:

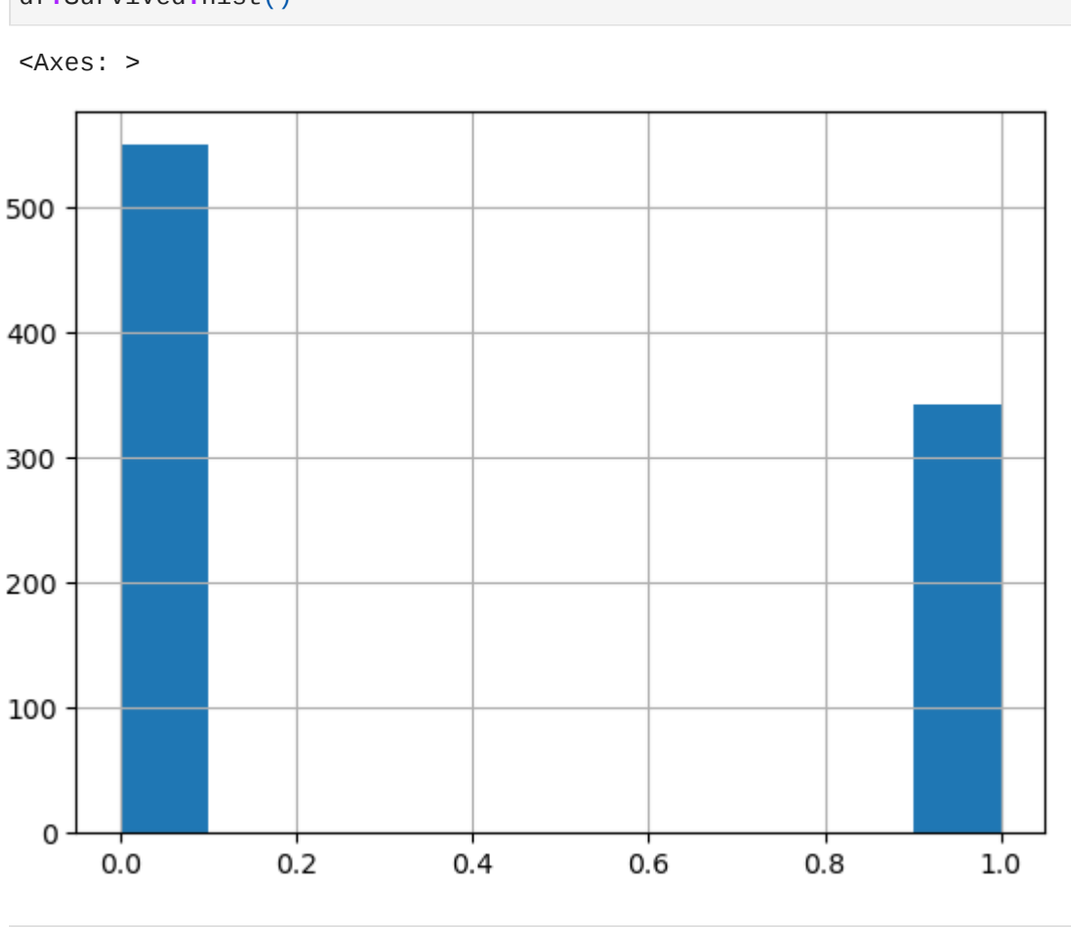
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	passenger_title
0	1	0	3	male	22.0	1	0	A/5 21171	7.2500	S	Mr
1	2	1	1	female	38.0	1	0	PC 17599	71.2833	C	Mrs
2	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	S	Miss
3	4	1	1	female	35.0	1	0	113803	53.1000	S	Mrs
4	5	0	3	male	35.0	0	0	373450	8.0500	S	Mr

```
In [10]: cols=list(df.columns.values)
cols.pop(cols.index('Survived'))
df=df[cols+['Survived']]
df.Survived.value_counts()

Out[10]: 0    549
1    342
Name: Survived, dtype: int64
```

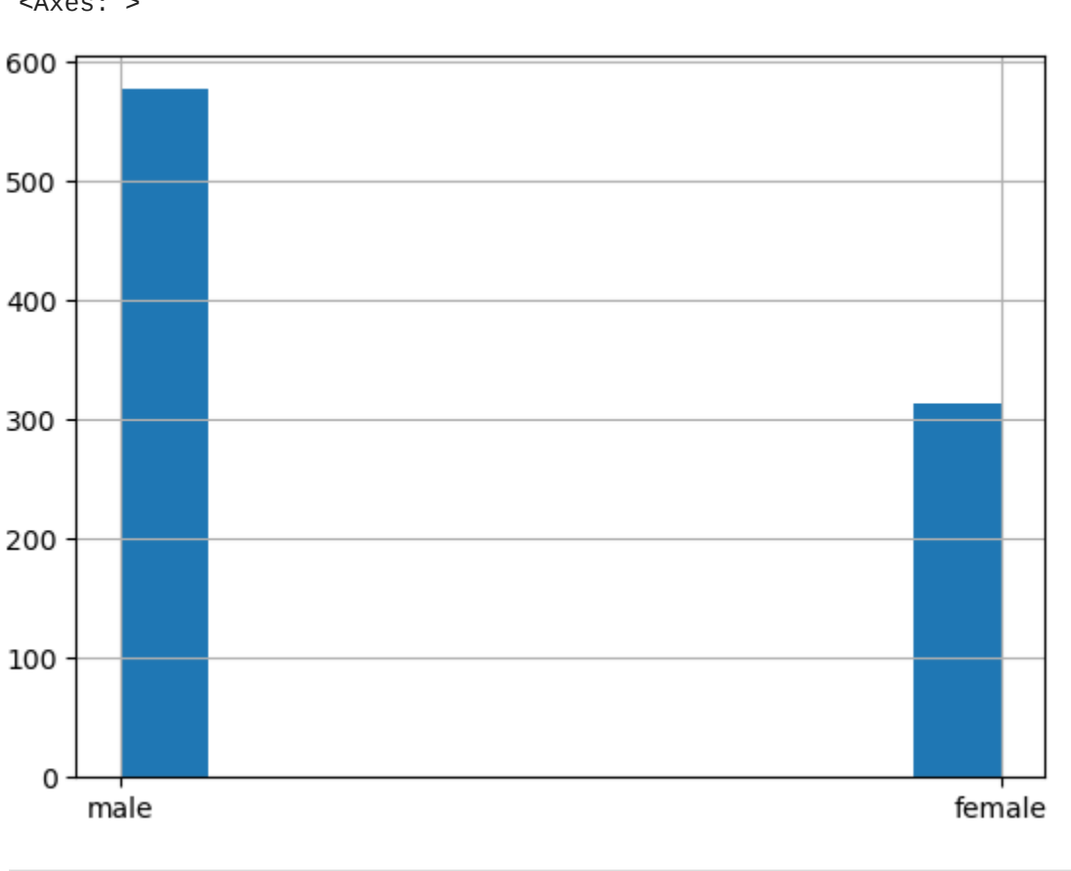
```
In [11]: df.Sex.value_counts()

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

Data Visualization

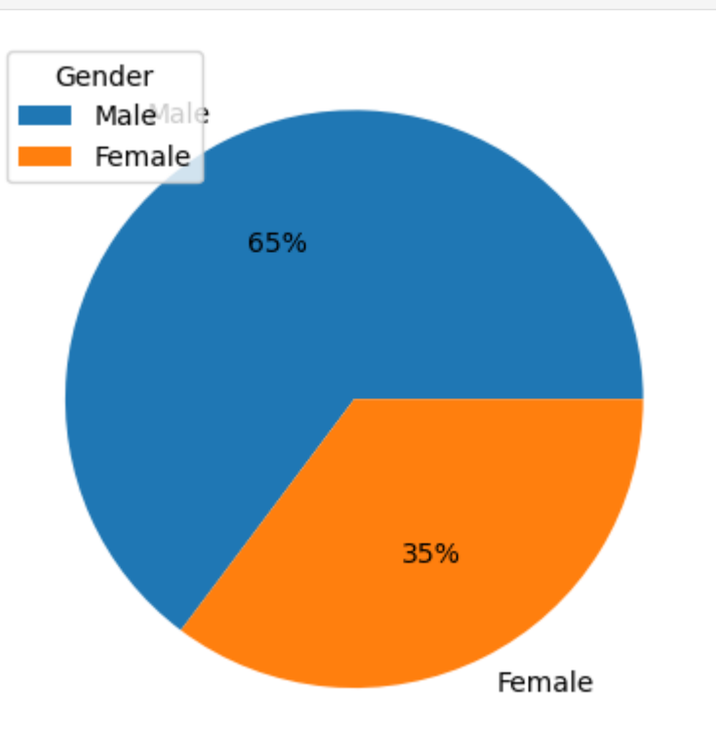
```
In [15]: df.Survived.hist()

Out[15]: <Axes: >
```



```
In [16]: df.Sex.hist()

Out[16]: <Axes: >
```

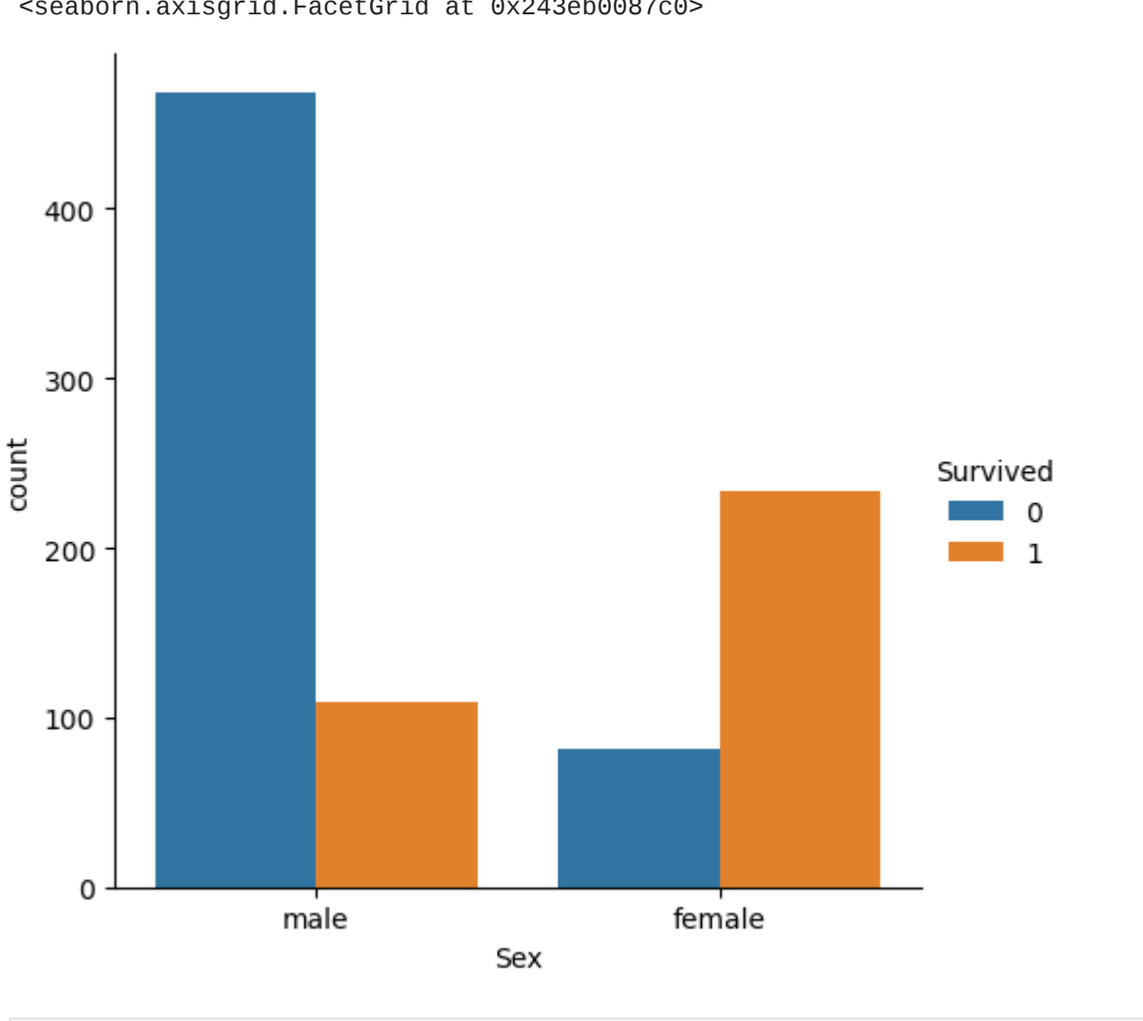



```
In [18]: gender=df.Sex.value_counts()
gender_labels=['Male','Female']
plt.pie(gender,labels=gender_labels,autopct='%.0f%%')
plt.legend(title='Gender')
plt.show()
```

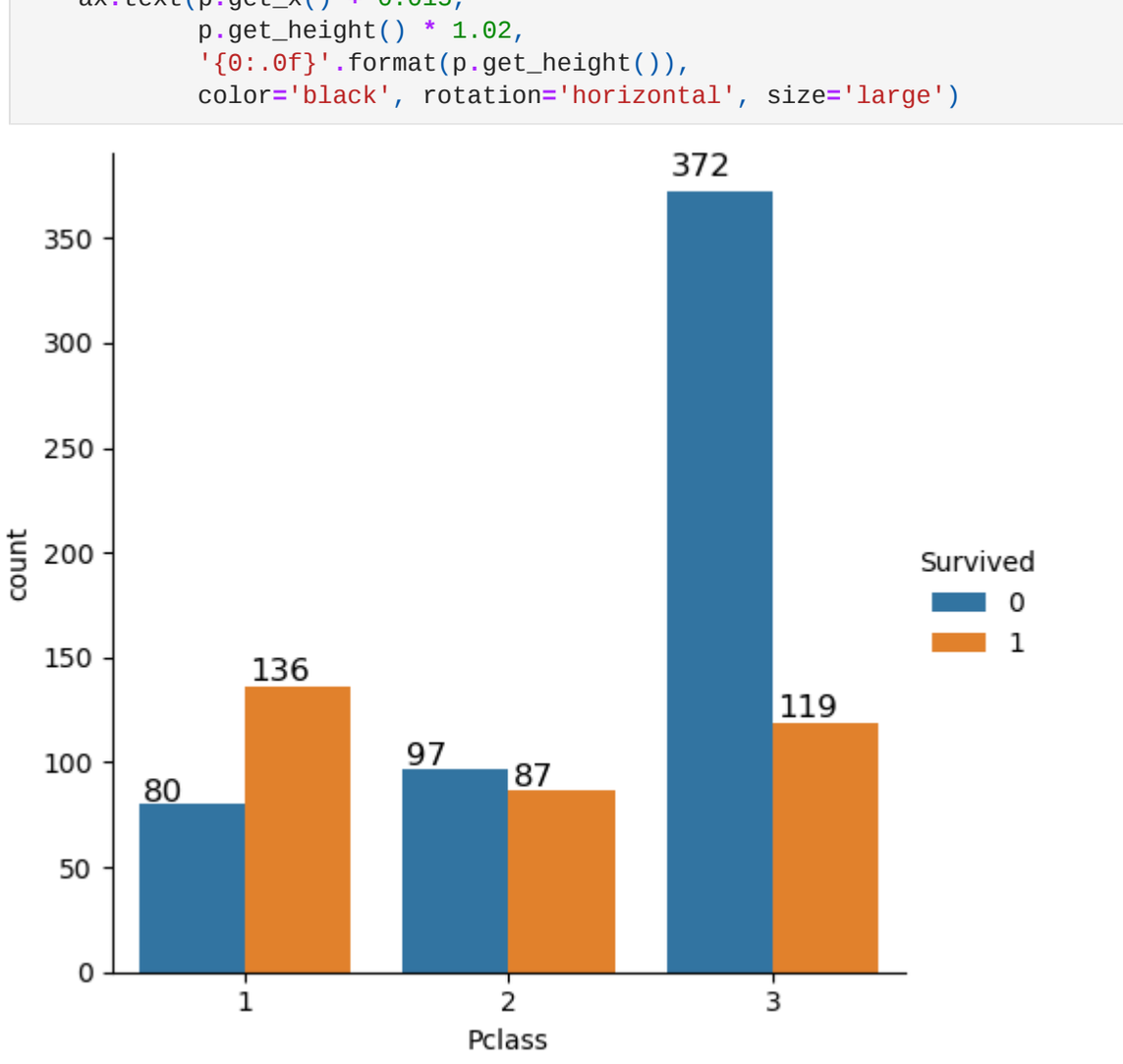


Passenger survival across male and female

```
In [19]: sns.catplot(data=df,x='Sex',kind='count',hue='Survived')

Out[19]: <seaborn.axisgrid.FacetGrid at 0x243eb0087c0>
```

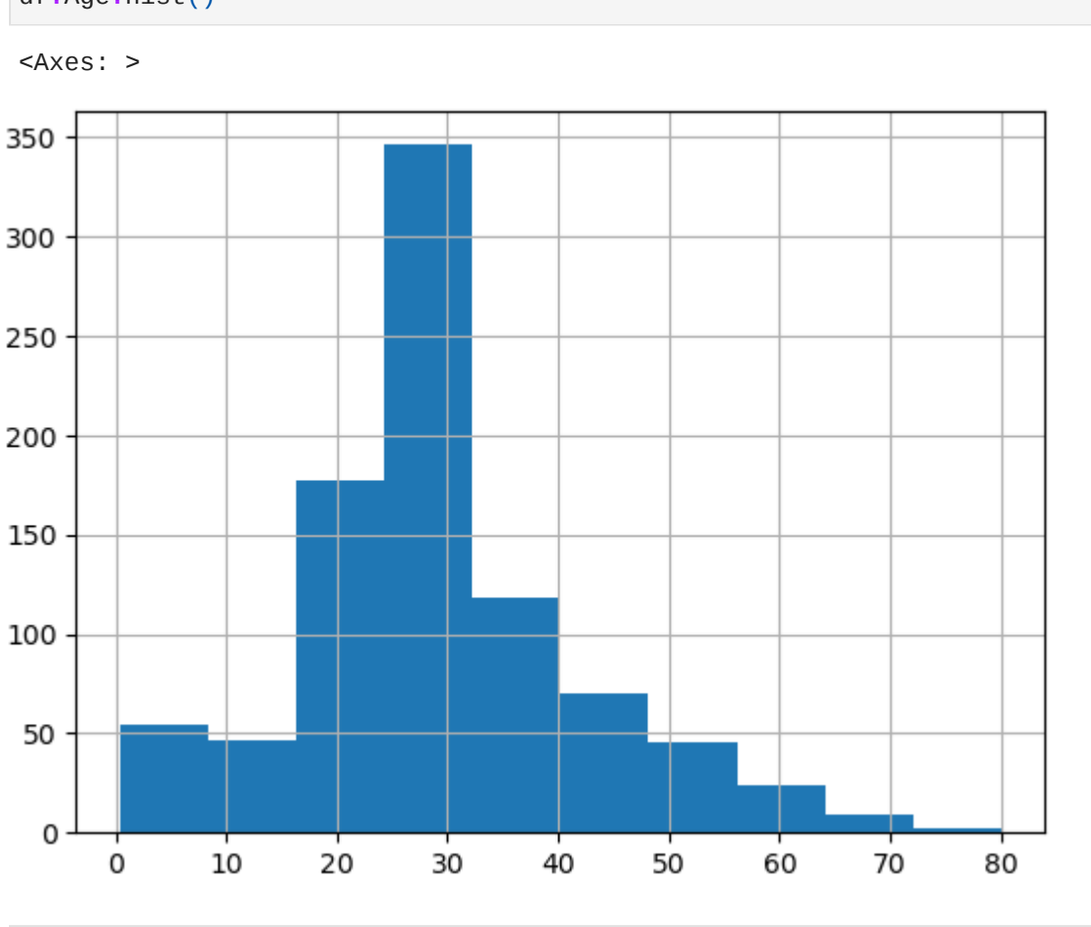



```
In [20]: #Passenger class survival
f=sns.catplot(data=df,x='Pclass',kind='count',hue='Survived')
ax = f.facet_axis(0,0)
for p in ax.patches:
    ax.text(p.get_x() + 0.015,
            p.get_height() * 1.02,
            '{0:.0f}'.format(p.get_height()),
            color='black', rotation='horizontal', size='large')
```



Passenger Age distribution

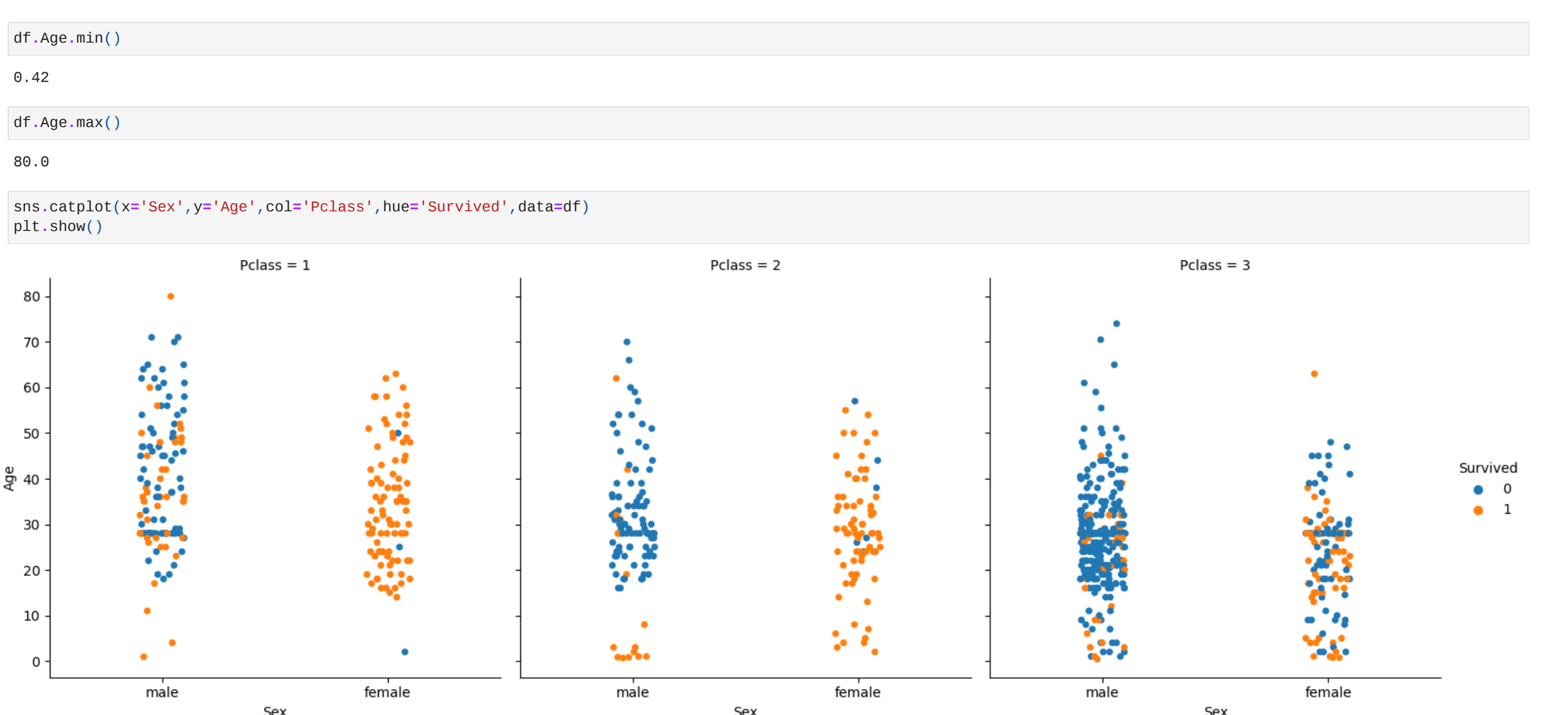
```
In [21]: df.Age.hist()

Out[21]: <Axes: >
```



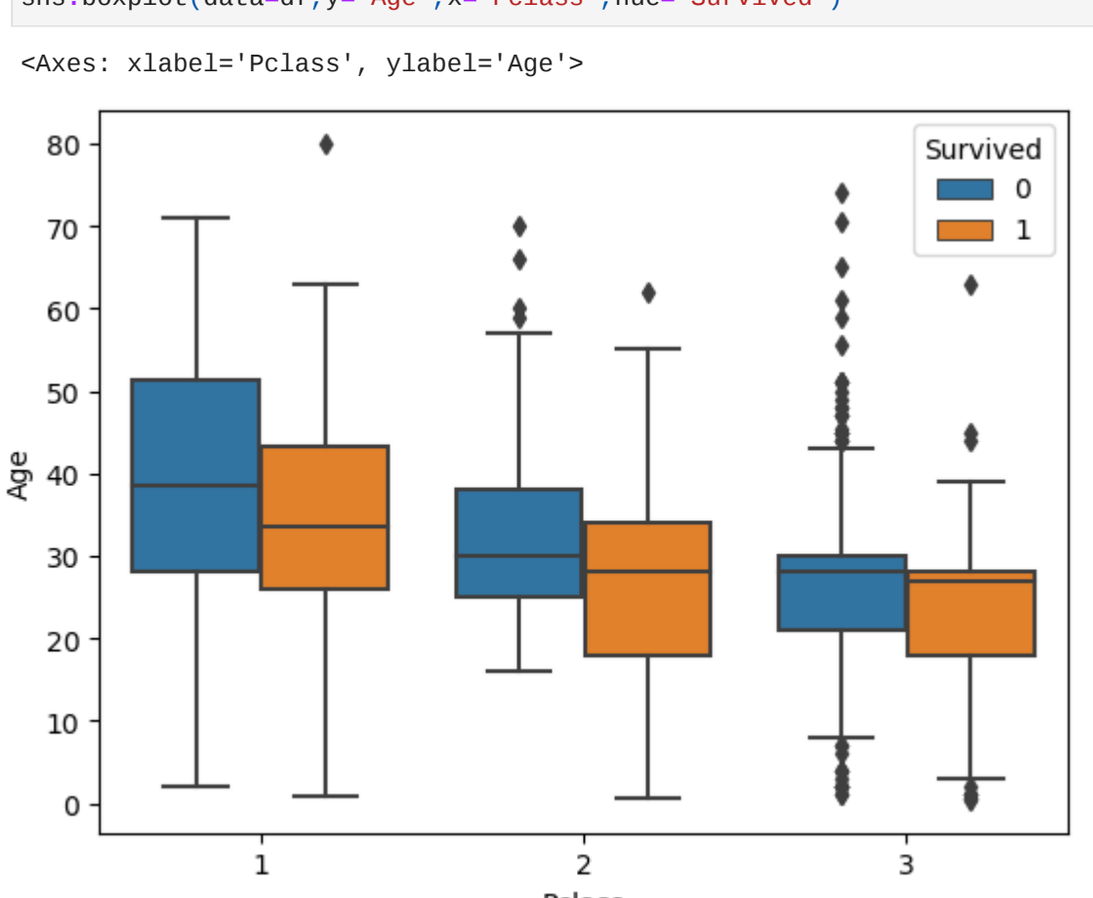
```
In [22]: df.Age.min()

Out[22]: 0.42
```

```
In [23]: df.Age.max()

Out[23]: 80.0
```

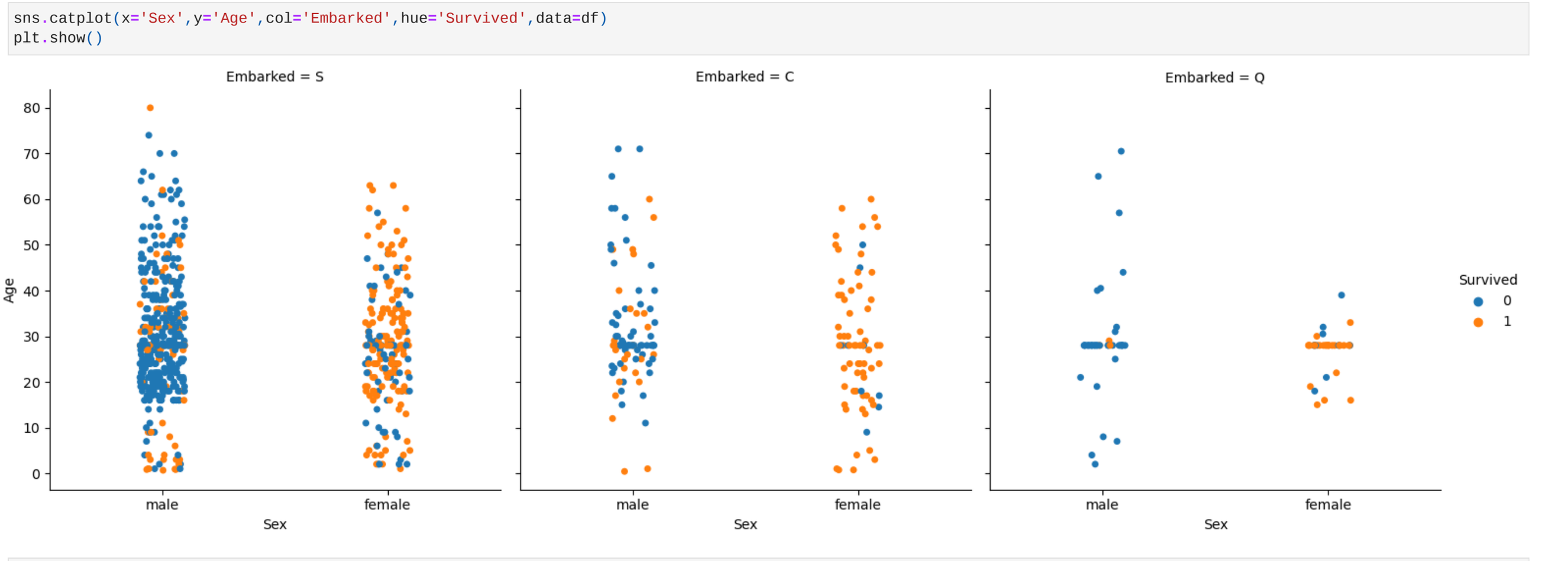
```
In [24]: sns.catplot(x='Sex',y='Age',col='Pclass',hue='Survived',data=df)
plt.show()
```



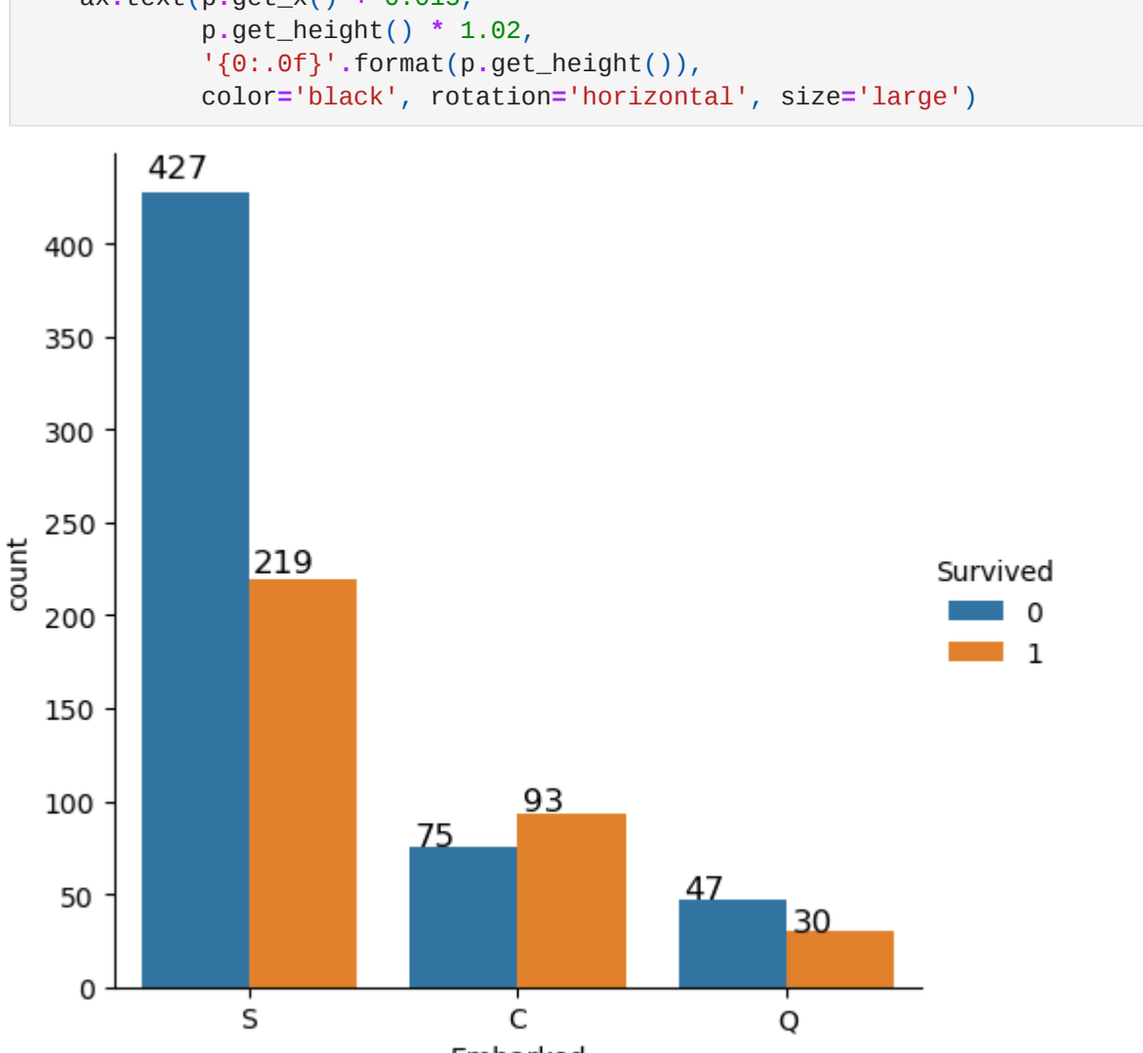

```
In [25]: sns.boxplot(data=df,y='Age',x='Pclass',hue='Survived')

Out[25]: <Axes: xlabel='Pclass', ylabel='Age'>
```



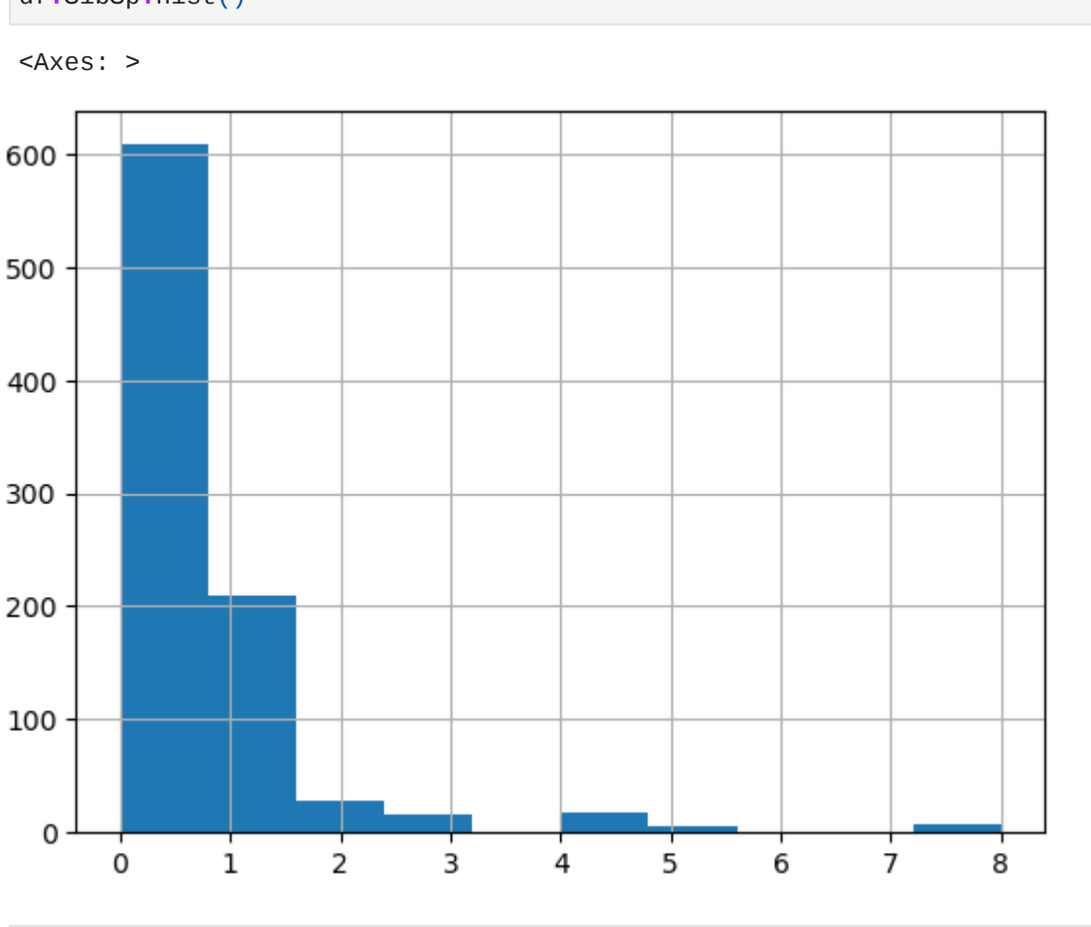

```
In [26]: sns.catplot(x='Sex',y='Age',col='Embarked',hue='Survived',data=df)
plt.show()
```




```
In [27]: #Passenger survival across differend embarkment
g=sns.catplot(data=df,x='Embarked',kind='count',hue='Survived')
ax = g.facet_axis(0,0)
for p in ax.patches:
    ax.text(p.get_x() + 0.015,
            p.get_height() * 1.02,
            '{0:.0f}'.format(p.get_height()),
            color='black', rotation='horizontal', size='large')
```

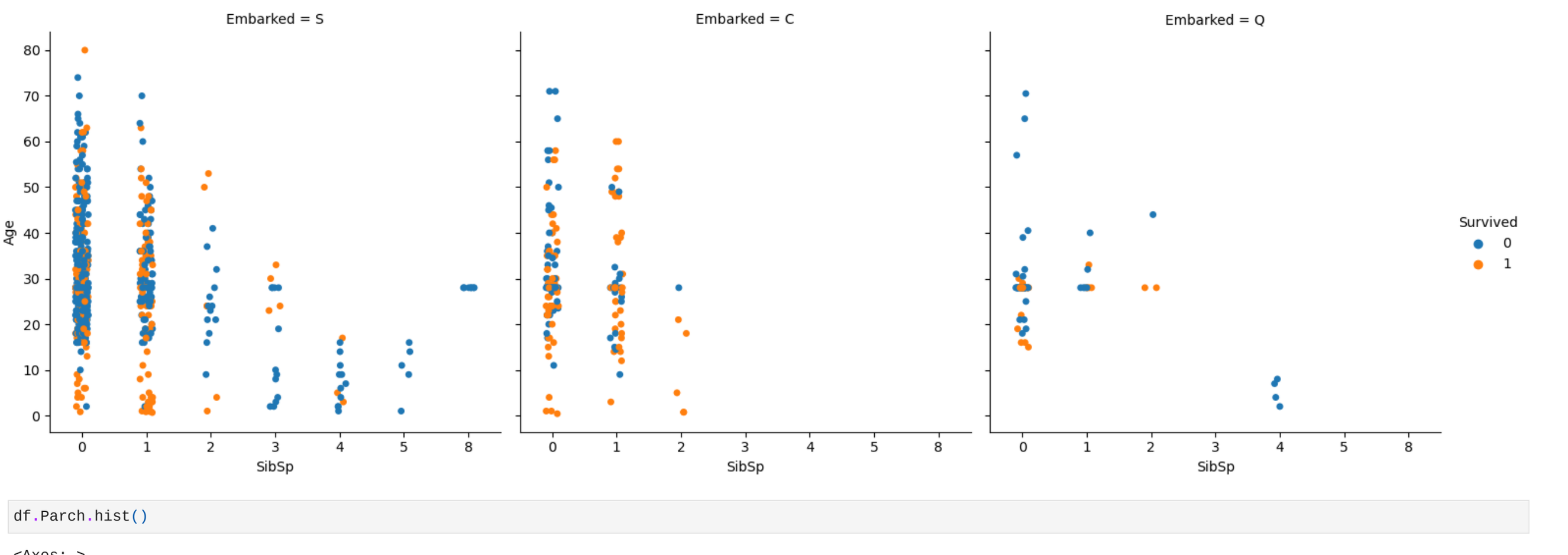



```
In [28]: df.SibSp.hist()

Out[28]: <Axes: >
```

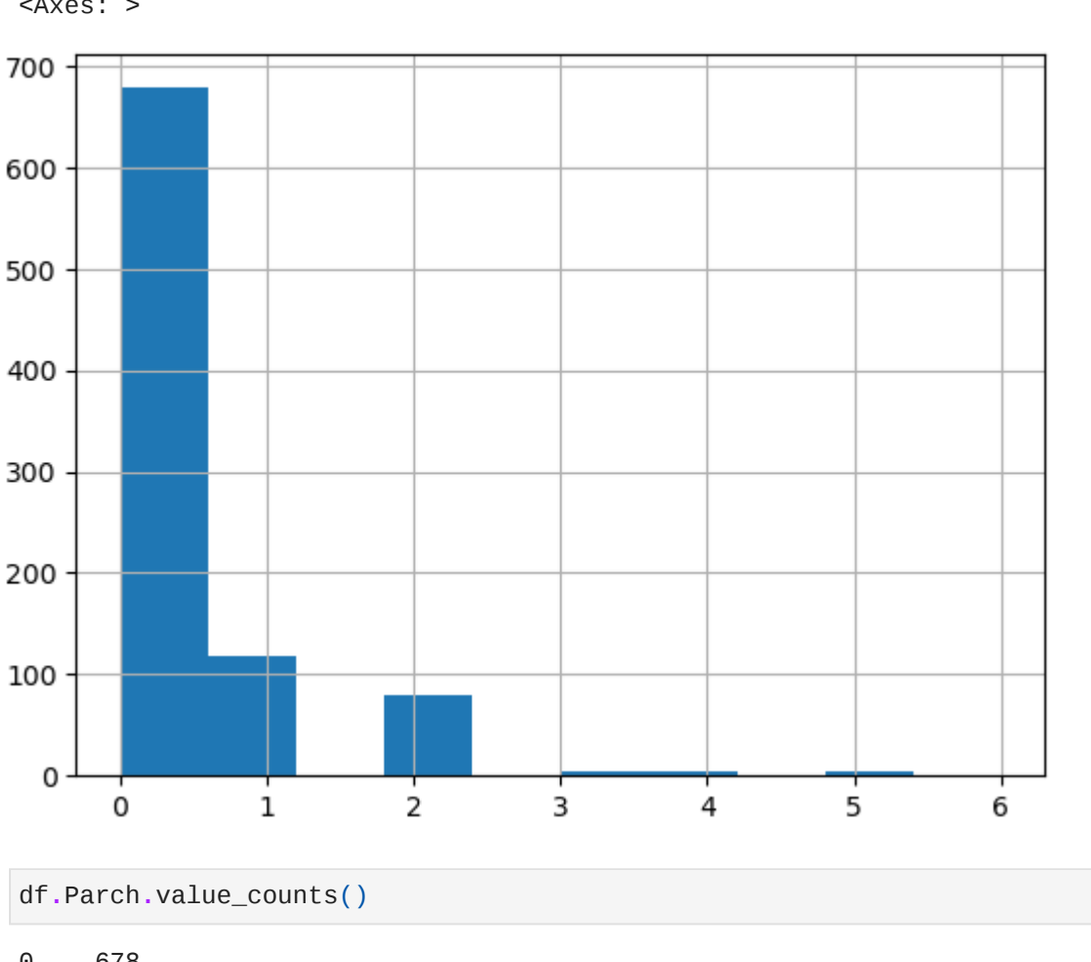


```
In [29]: df.SibSp.value_counts()

Out[29]: 0    608
1    209
2     28
4     18
3     16
8      7
5      5
Name: SibSp, dtype: int64
```

```
In [30]: sns.catplot(x='SibSp',y='Age',col='Embarked',hue='Survived',data=df)
plt.show()
```



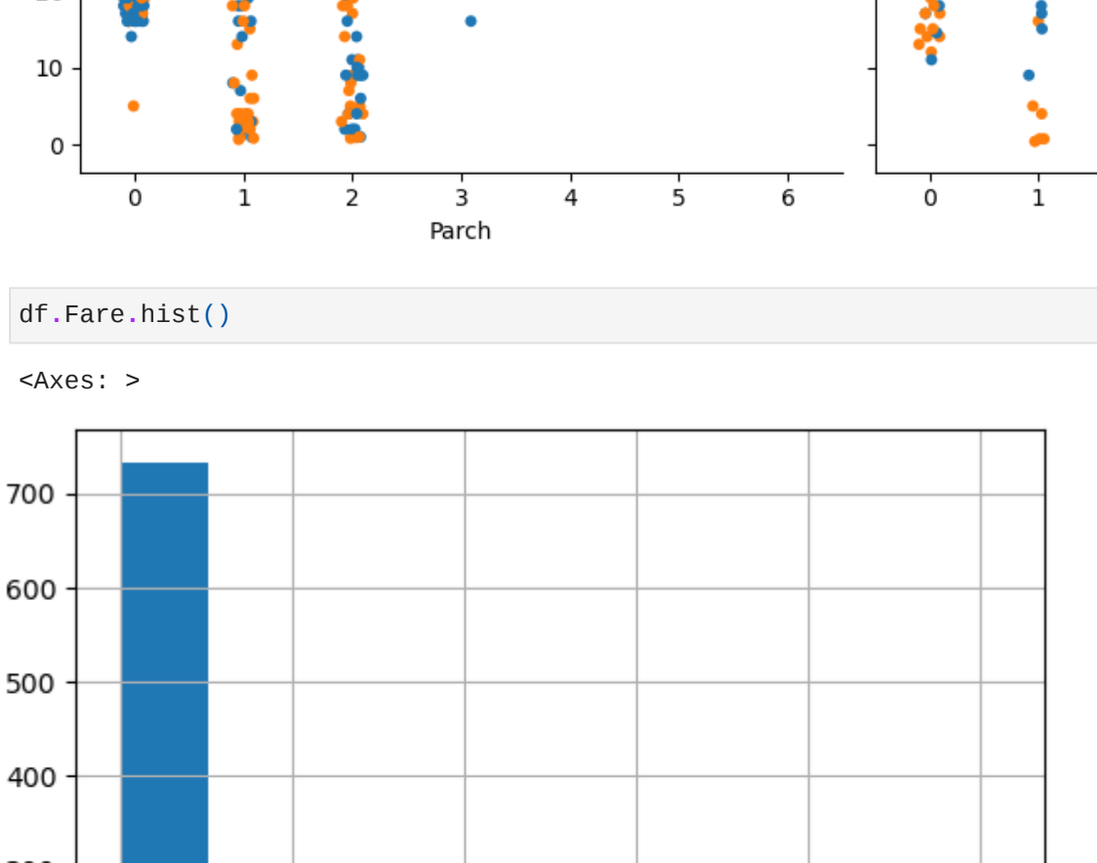

```
In [31]: df.Parch.hist()

Out[31]: <Axes: >
```



```
In [32]: df.Parch.value_counts()

Out[32]: 0    678
1    118
2     80
5      5
3      5
4      4
6      1
Name: Parch, dtype: int64
```

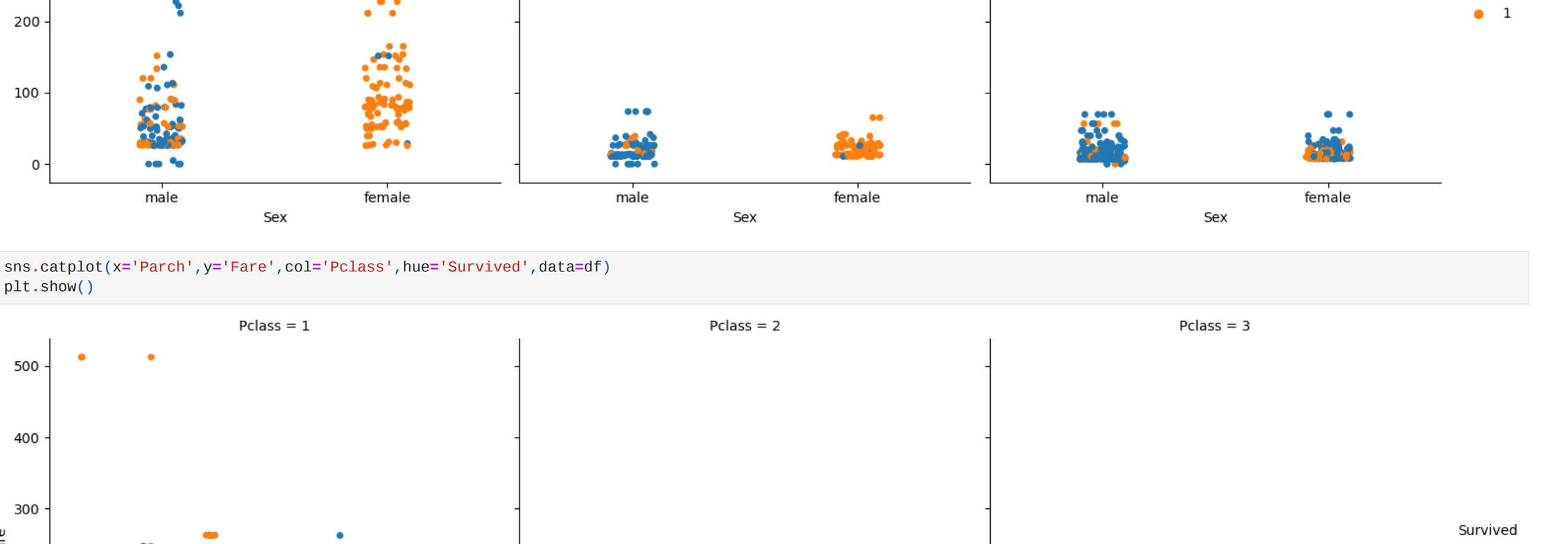
```
In [33]: sns.catplot(x='Parch',y='Age',col='Embarked',hue='Survived',data=df)
plt.show()
```


```
In [34]: df.Fare.hist()

Out[34]: <Axes: >
```

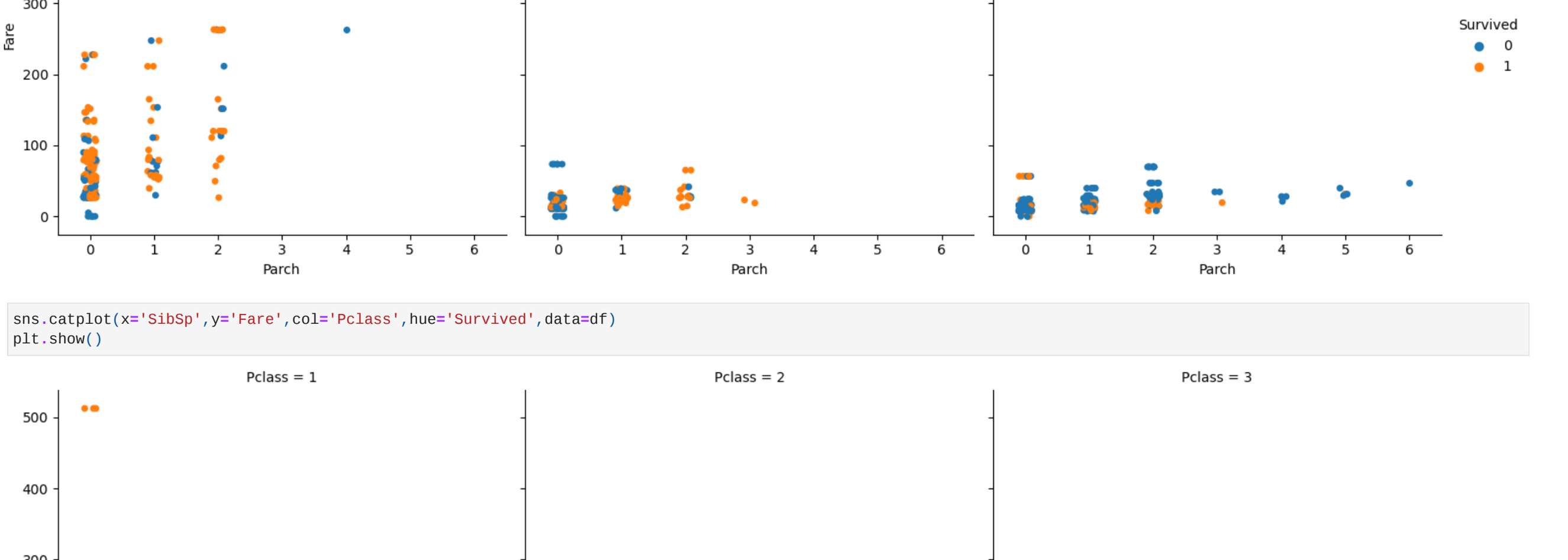


```
In [35]: sns.catplot(x='Sex',y='Fare',col='Pclass',hue='Survived',data=df)
plt.show()
```


```
In [36]: sns.catplot(x='Parch',y='Fare',col='Pclass',hue='Survived',data=df)
plt.show()
```




```
In [37]: sns.catplot(x='SibSp',y='Fare',col='Pclass',hue='Survived',data=df)
plt.show()
```




```
In [39]: X=df.drop(['Fare','Embarked','passenger_title','Ticket','Survived','PassengerId'],axis=1)
y=df.Survived
```

```
In [40]: from sklearn.preprocessing import LabelEncoder
lbl_enc=LabelEncoder()
X['Sex']=lbl_enc.fit_transform(X['Sex'])
```

```
In [41]: np.random.state=42
train_X,test_X,train_y,test_y=train_test_split(X,y,test_size=0.2,random_state=42)
```

Logistic Regression

```
In [45]: log_reg=LogisticRegression()
log_reg.fit(train_X,train_y)
pred=log_reg.predict(test_X)
train=log_reg.predict(train_X)
round(accuracy_score(test_y,pred),4)

Out[45]: 0.8156
```

Decision Tree Classifier

```
In [46]: dt_model=DecisionTreeClassifier()
dt_model.fit(train_X,train_y)
dt_pred=dt_model.predict(test_X)
dt_train=dt_model.predict(train_X)
round(accuracy_score(test_y,dt_pred),4)

Out[46]: 0.7654
```

Random Forest Classifier

```
In [47]: rf_model=RandomForestClassifier()
rf_model.fit(train_X,train_y)
rf_pred=rf_model.predict(test_X)
rf_train=rf_model.predict(train_X)
round(accuracy_score(test_y,rf_pred),4)

Out[47]: 0.8212
```

From the scores of our models Randomforest performed with better results followed by logistic regression then Decision Trees.