EDA to Prediction (DieTanic)

Predictive Modeling on the Titanic Dataset

EunSeo Ko
2024.06.18.

Introduction

The Titanic Disaster

The Titanic sank on April 15, 1912, after hitting an iceberg.

Out of 2224 passengers and crew, 1502 lost their lives.

The Titanic's sinking remains one of the most infamous shipwrecks in history.

Specification

Exploratory Data Analysis (EDA)

Analysis of individual features.

Exploring relationships between multiple features.

Identifying trends and patterns.

Feature Engineering and Data Cleaning

Adding new features to enhance the dataset.

Removing redundant or unnecessary features.

Converting features into suitable forms for modeling.

Predictive Modeling

Running basic algorithms to build models.

Performing cross-validation to ensure model reliability.

Using ensembling techniques to improve accuracy.

Extracting the most important features.

Codes

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [3]:

```
data=pd.read_csv('/content/sample_data/train.csv')
```

In [4]:

```
data.head()
```

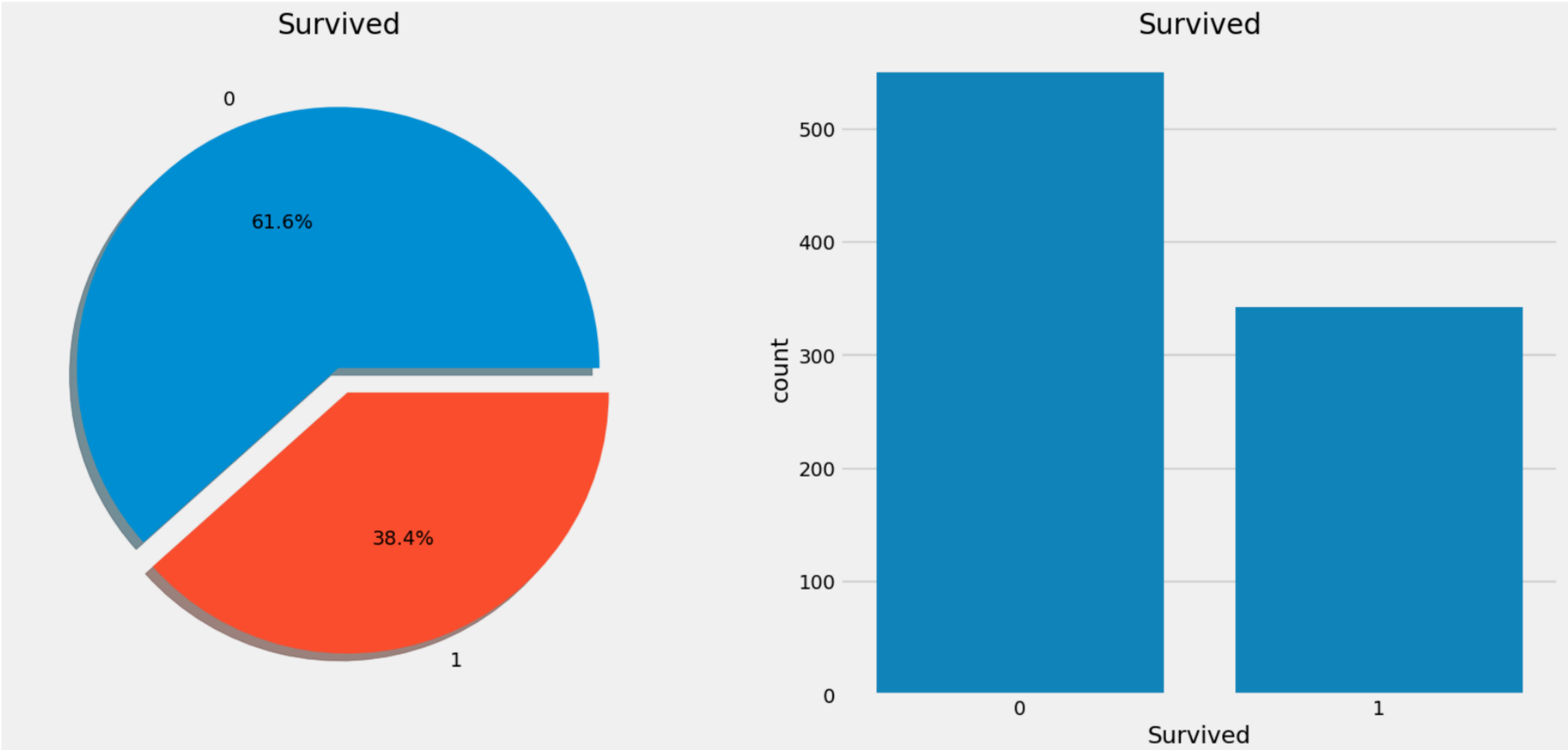
Out[4]:

	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

In [7]:

```
f,ax=plt.subplots(1,2,figsize=(18,8))
data['Survived'].value_counts().plot.pie(explode=[0,0.1],autopct='%1.1f%%',ax=ax[0],shadow=True)
ax[0].set_title('Survived')
ax[0].set_ylabel('')
sns.countplot(x = 'Survived',data=data,ax=ax[1]) # x = 추가
ax[1].set_title('Survived')
plt.show()
```

File display

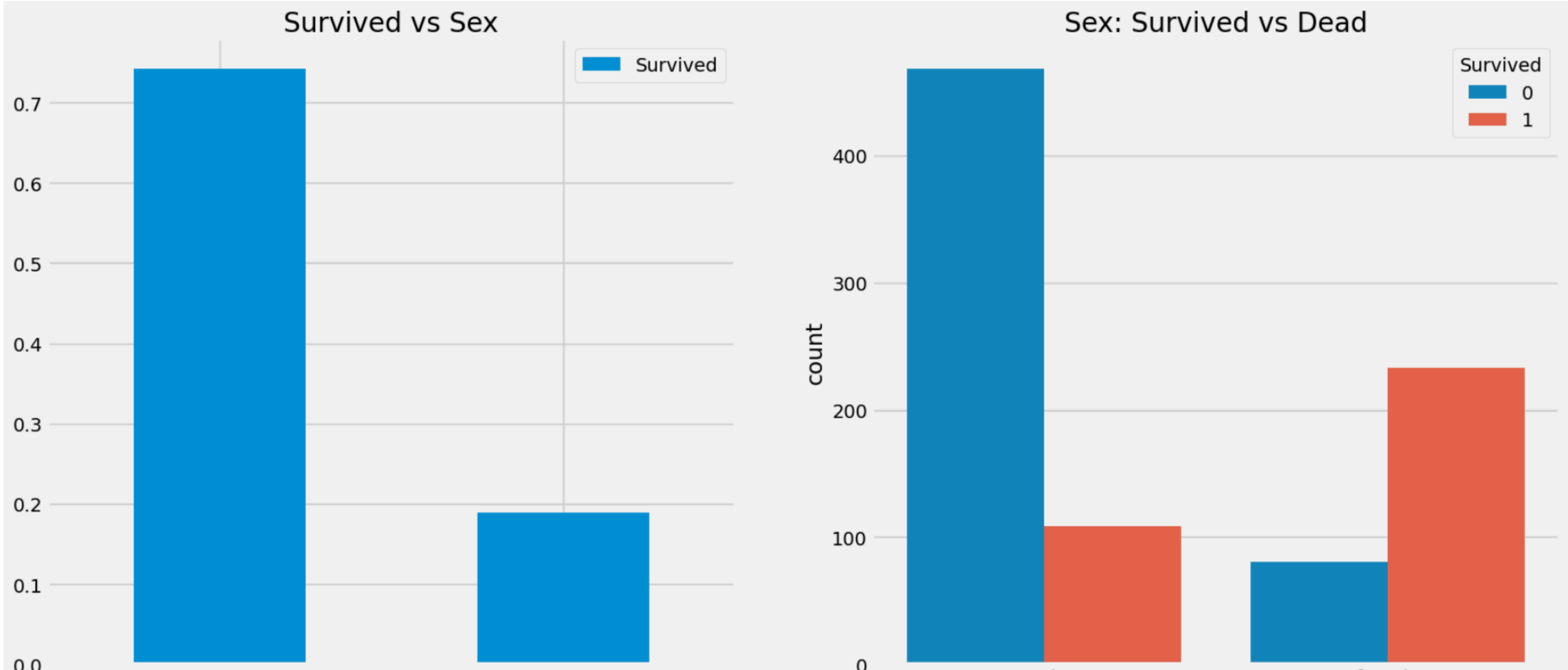


```python
# 첫 번째 서브플롯: Sex별 생존율 평균
data[['Sex', 'Survived']].groupby(['Sex']).mean().plot.bar(ax=ax[0])
ax[0].set_title('Survived vs Sex')

# 두 번째 서브플롯: Sex별 생존자와 사망자 수
sns.countplot(x='Sex', hue='Survived', data=data, ax=ax[1])
ax[1].set_title('Sex: Survived vs Dead')

# 그래프 출력
plt.show()
```

File display

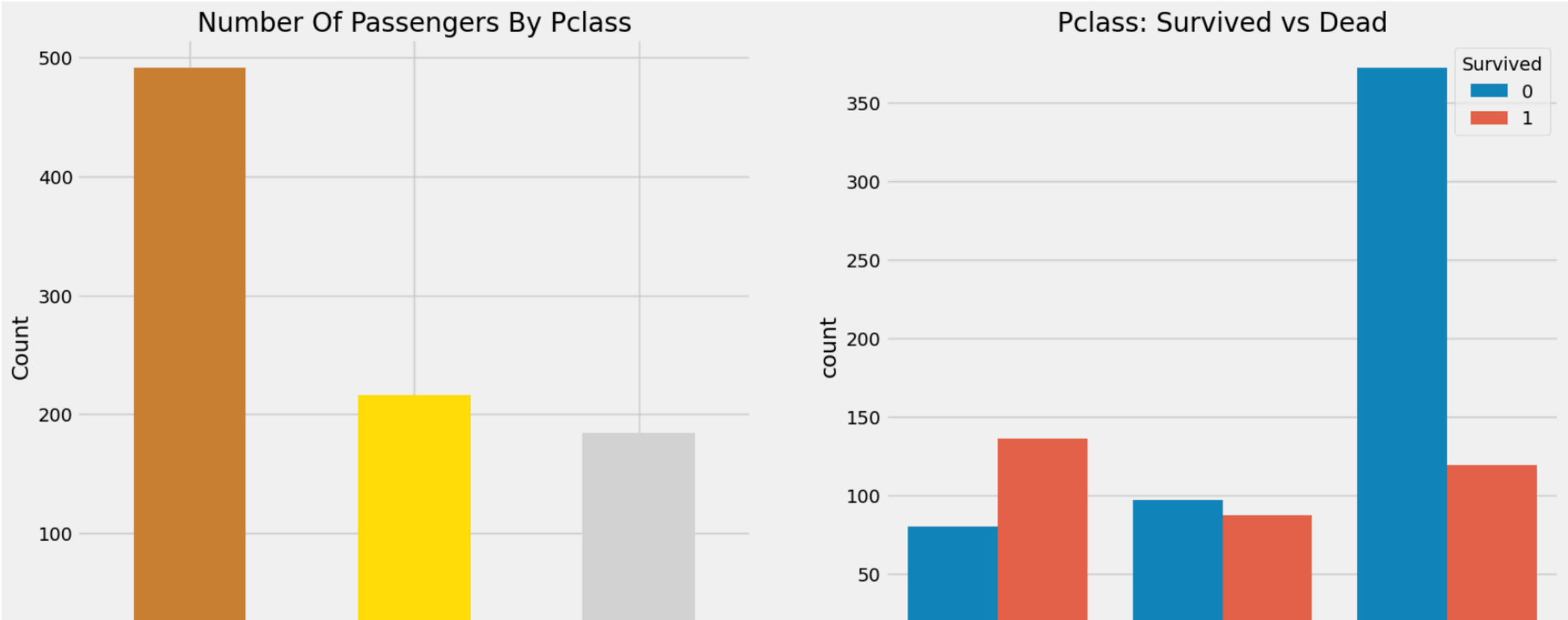


```
# 첫 번째 서브플롯: Pclass별 승객 수
data['Pclass'].value_counts().plot.bar(color=['#CD  FFDF00', '#D3D3D3'], ax=ax[0])
ax[0].set_title('Number Of Passengers By Pclass')
ax[0].set_ylabel('Count')

# 두 번째 서브플롯: Pclass별 생존자와 사망자 수
sns.countplot(x='Pclass', hue='Survived', data=data, ax=ax[1])
ax[1].set_title('Pclass: Survived vs Dead')

# 그래프 출력
plt.show()
```

File display

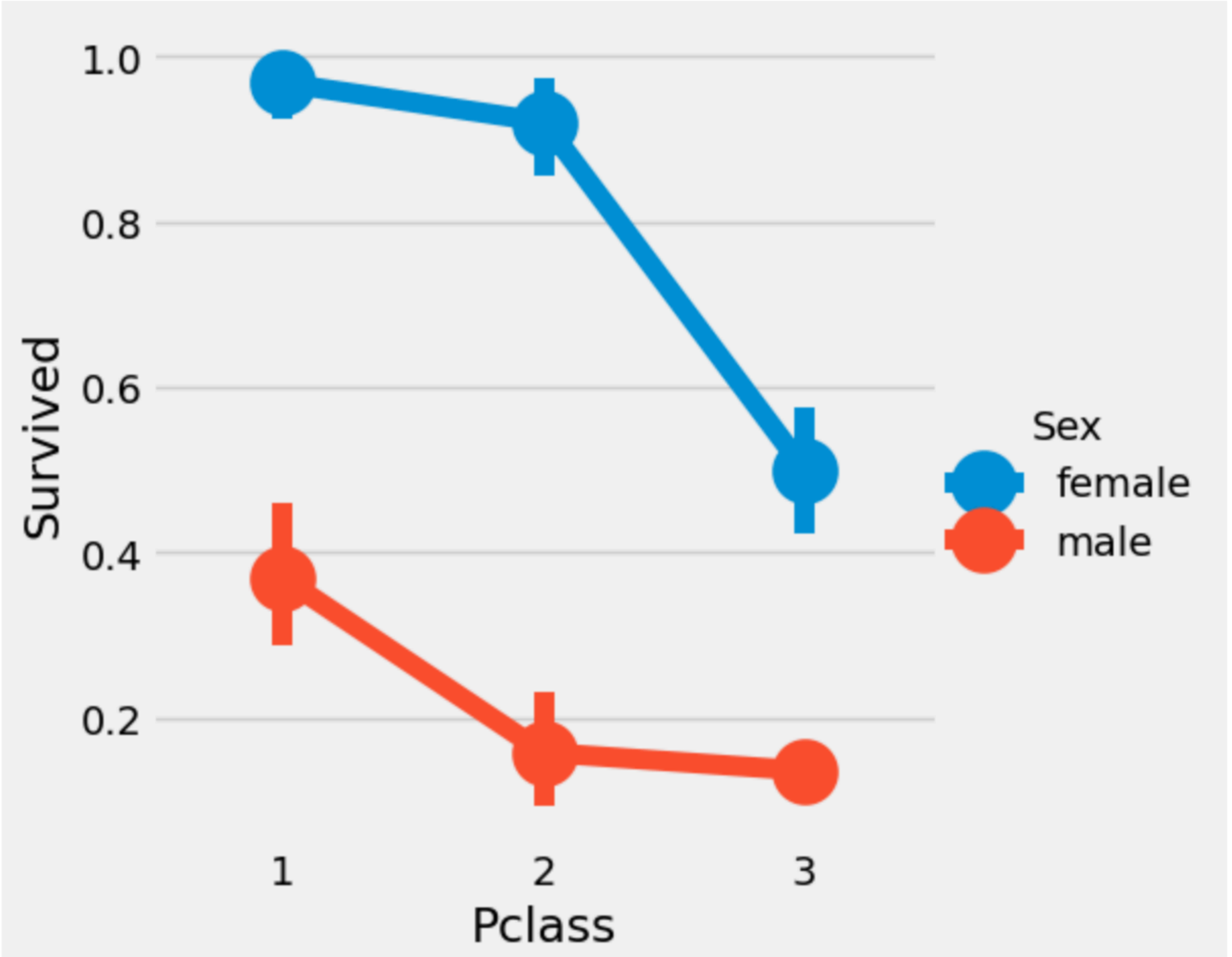


In [16]:

```
# catplot을 사용하여 Pclass, Survived, Sex에 따른 그래프 생성
sns.catplot(x='Pclass', y='Survived', hue='Sex', data=data, kind='point')

# 그래프 출력
plt.show()
```

File display



```
# 첫 번째 서브플롯: Pclass와 Age에 따른 생존 여부
sns.violinplot(x="Pclass", y="Age", hue="Survived", data=data, split=True, ax=ax[0])
ax[0].set_title('Pclass and Age vs Survived')
ax[0].set_yticks(range(0, 110, 10))

# 두 번째 서브플롯: Sex와 Age에 따른 생존 여부
sns.violinplot(x="Sex", y="Age", hue="Survived", data=data, split=True, ax=ax[1])
ax[1].set_title('Sex and Age vs Survived')
ax[1].set_yticks(range(0, 110, 10))

# 그래프 출력
plt.show()
```

```
data['Initial']=0
for i in data:
    data['Initial']=data.Name.str.extract('([A-Za-z]+)\.') #lets extract the Salutations
```

In [21]:
```
pd.crosstab(data.Initial,data.Sex).T.style.background_gradient(cmap='summer_r') #Checking the Initials with the S
```

Out[21]:

Initial	Capt	Col	Countess	Don	Dr	Jonkheer	Lady	Major	Master	Miss	Mlle	Mme	Mr	Mrs	Ms	Rev	Sir
Sex																	
female	0	0	1	0	1	0	1	0	0	182	2	1	0	125	1	0	0
male	1	2	0	1	6	1	0	2	40	0	0	0	517	0	0	6	1

In [22]:
```
data['Initial'].replace(['Mlle','Mme','Ms','Dr','Major','Lady','Countess','Jonkheer','Col','Rev','Capt','Sir','Do
```

In [23]:
```
data.groupby('Initial')['Age'].mean() #lets check the average age by Initials
```

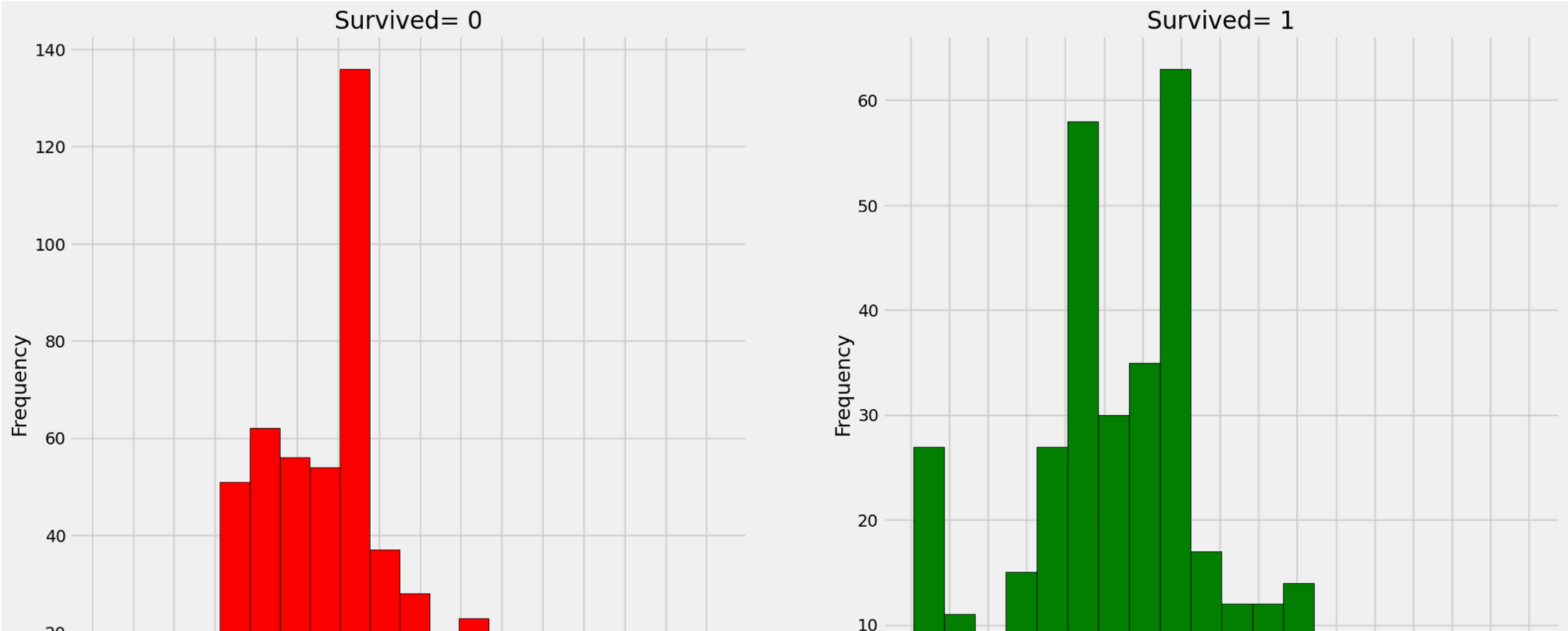
Out[23]:
```
Initial
Master     4.574167
Miss      21.860000
Mr        32.739609
Mrs       35.981818
Other     45.888889
Name: Age, dtype: float64
```

In [24]:
```
## Assigning the NaN Values with the Ceil values of the mean ages
data.loc[(data.Age.isnull())&(data.Initial=='Mr'),'Age']=33
data.loc[(data.Age.isnull())&(data.Initial=='Mrs'),'Age']=36
data.loc[(data.Age.isnull())&(data.Initial=='Master'),'Age']=5
data.loc[(data.Age.isnull())&(data.Initial=='Miss'),'Age']=22
data.loc[(data.Age.isnull())&(data.Initial=='Other'),'Age']=46
```

File display

In [26]:

```
f,ax=plt.subplots(1,2,figsize=(20,10))
data[data['Survived']==0].Age.plot.hist(ax=ax[0],bins=20,edgecolor='black',color='red')
ax[0].set_title('Survived= 0')
x1=list(range(0,85,5))
ax[0].set_xticks(x1)
data[data['Survived']==1].Age.plot.hist(ax=ax[1],color='green',bins=20,edgecolor='black')
ax[1].set_title('Survived= 1')
x2=list(range(0,85,5))
ax[1].set_xticks(x2)
plt.show()
```

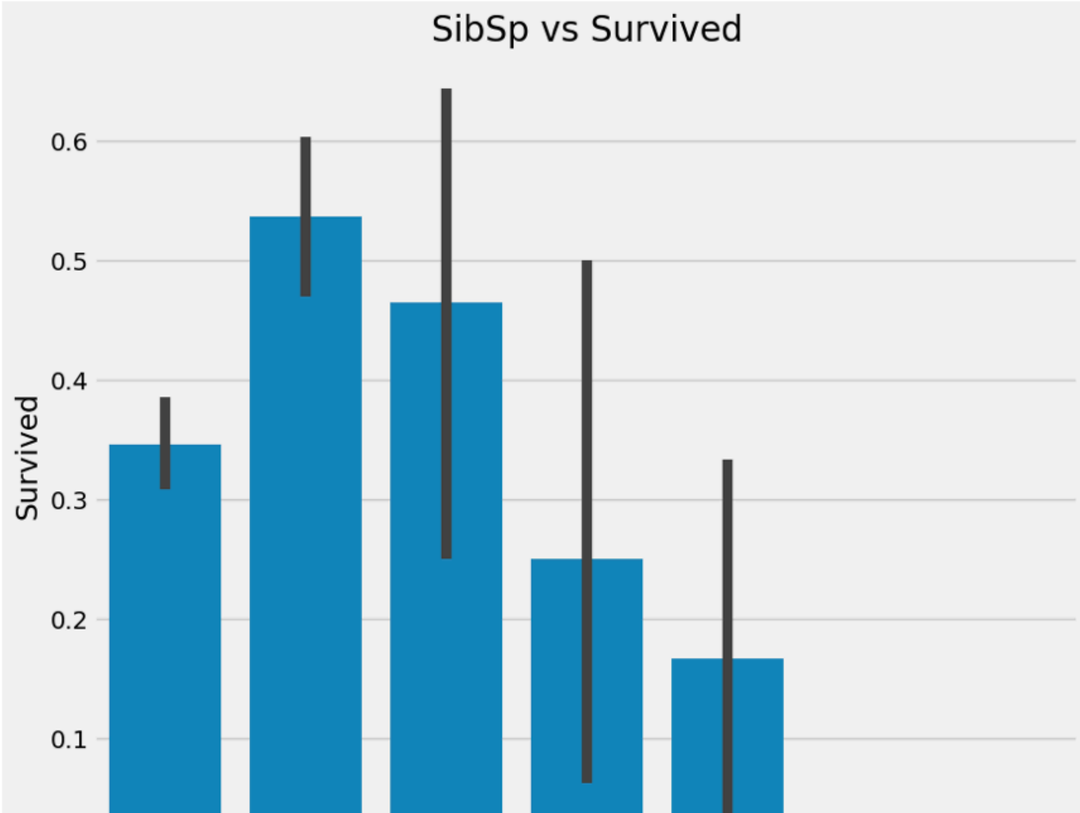


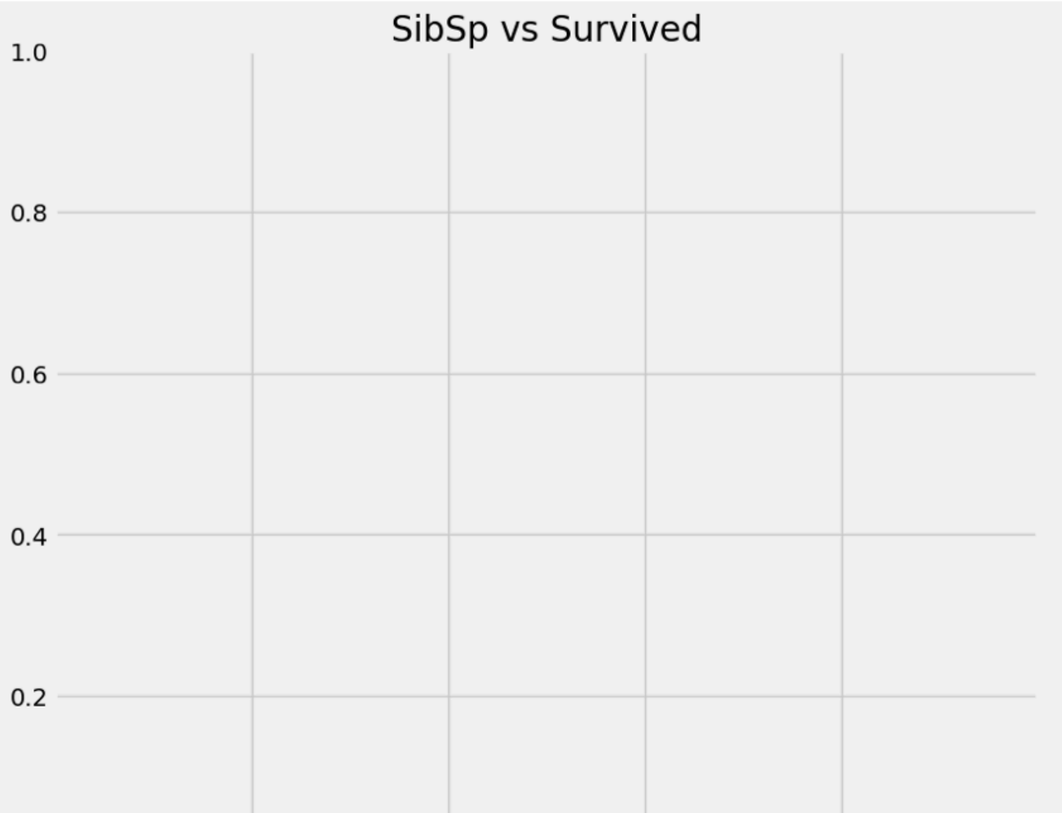
```
# 첫 번째 서브플롯: SibSp vs Survived (barplot)
sns.barplot(x='SibSp', y='Survived', data=data, ax=ax[0])
ax[0].set_title('SibSp vs Survived')

# 두 번째 서브플롯: SibSp vs Survived (catplot)
sns.catplot(x='SibSp', y='Survived', data=data, kind='bar', ax=ax[1])
ax[1].set_title('SibSp vs Survived')

# 두 번째 서브플롯의 catplot을 닫음
plt.close(2)

# 서브플롯 출력
plt.show()
```





Out[41]:

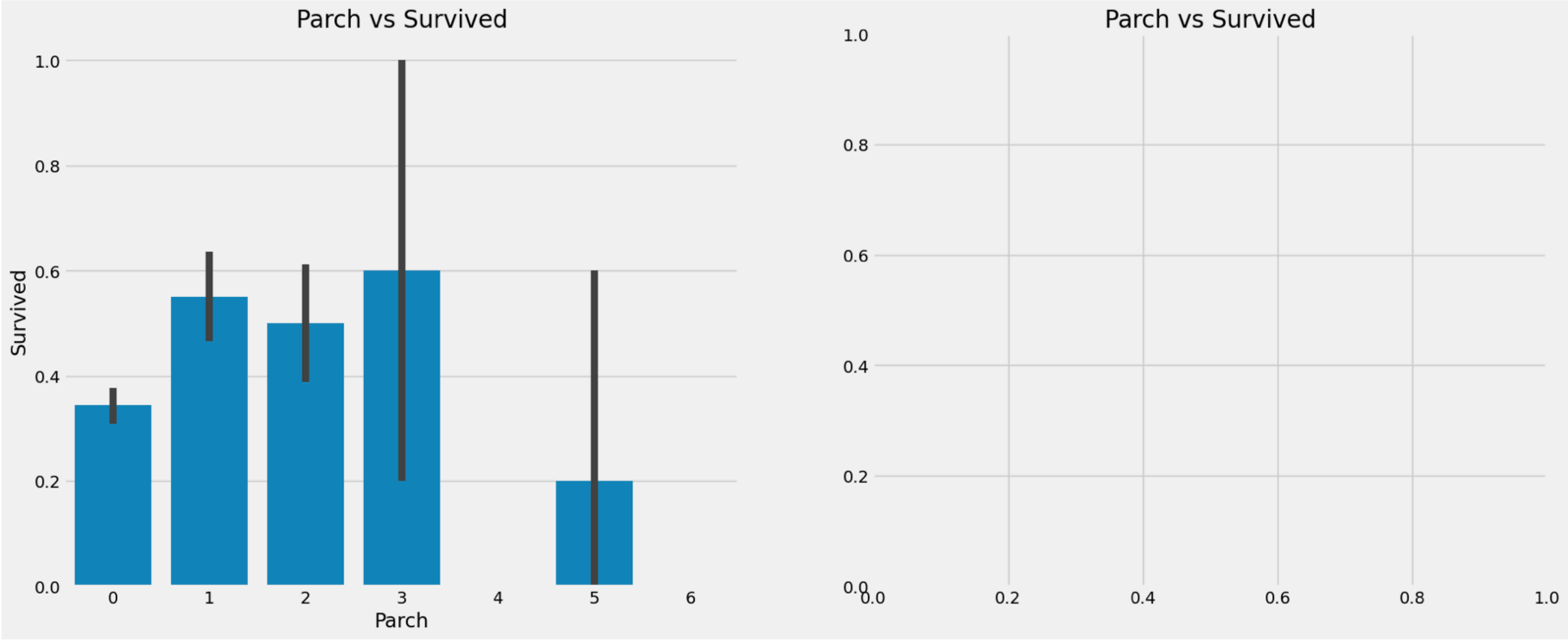
Pclass	1	2	3
SibSp			
0	137	120	351
1	71	55	83
2	5	8	15
3	3	1	12
4	0	0	18
5	0	0	5
8	0	0	7

In [42]:
```
pd.crosstab(data.Parch,data.Pclass).style.background_gradient(cmap='summer_r')
```

Out[42]:

Pclass	1	2	3
Parch			
0	163	134	381
1	31	32	55
2	21	16	43
3	0	2	3
4	1	0	3
5	0	0	5

```
ax[0].set_title('Parch vs Survived')
sns.catplot(x = 'Parch',y = 'Survived',data=data,ax=ax[1])
ax[1].set_title('Parch vs Survived')
plt.close(2)
plt.show()
```

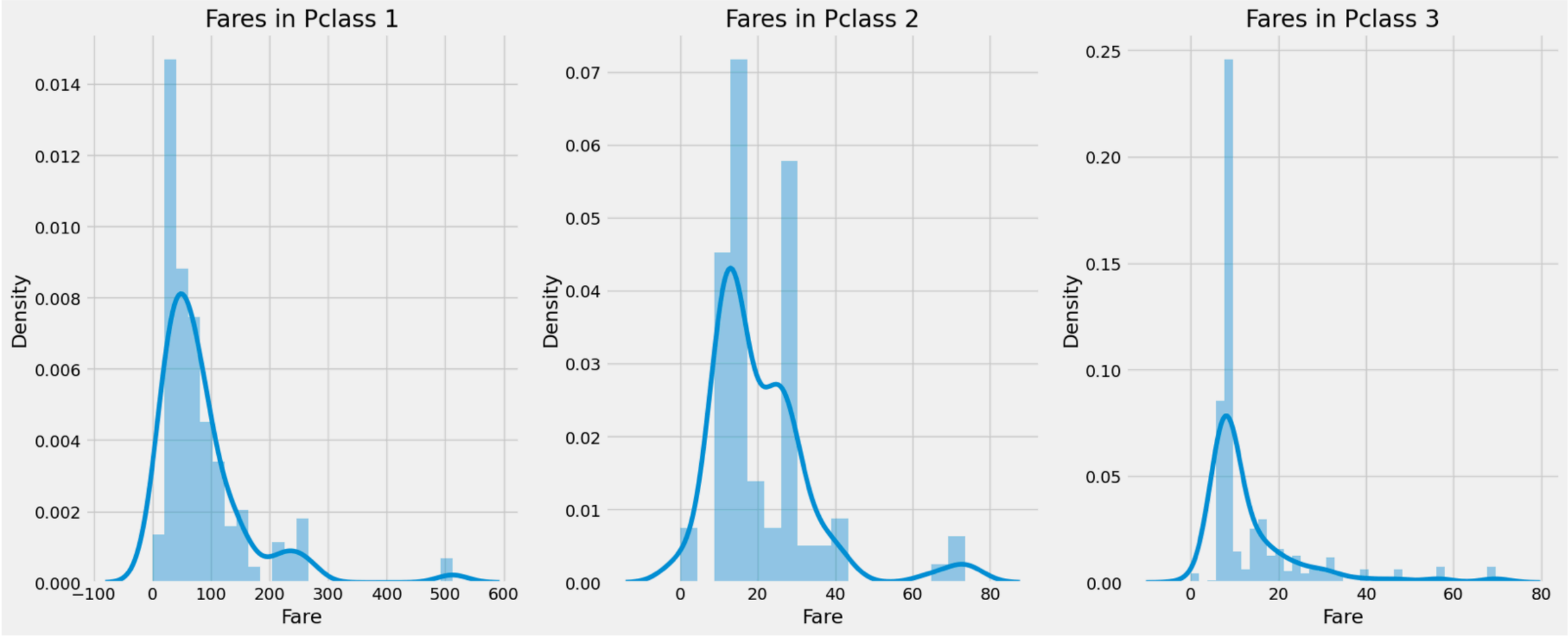


In [46]:

```
print('Highest Fare was:',data['Fare'].max())
print('Lowest Fare was:',data['Fare'].min())
print('Average Fare was:',data['Fare'].mean())
```

In [47]:

```
f,ax=plt.subplots(1,3,figsize=(20,8))
sns.distplot(data[data['Pclass']==1].Fare,ax=ax[0])
ax[0].set_title('Fares in Pclass 1')
sns.distplot(data[data['Pclass']==2].Fare,ax=ax[1])
ax[1].set_title('Fares in Pclass 2')
sns.distplot(data[data['Pclass']==3].Fare,ax=ax[2])
ax[2].set_title('Fares in Pclass 3')
plt.show()
```



Link to GitHub:

https://github.com/cannes7/DAStudy/blob/main/ESKDataAnalysis/DAStudy_kaggle_Titanic.ipynb

Q&A