19_ETL_Titanic

June 18, 2025

Creado por:

Isabel Maniega

1 Titanic Dataset - Predicción

Para competir en Kaggle será necesario descargar de esta página los csv de: train.csv, test.csv, gender_submission.csv

https://www.kaggle.com/competitions/titanic/data

[1]:
```
# pip install seaborn
```

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

Importamos el dataset train.csv

[3]:
```
df = pd.read_csv("./data/train.csv")
df.head()
```

[3]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
```

```
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

Borramos la columna de PassengerId

[4]:
```
df = df.drop("PassengerId", axis=1)
df
```

[4]:
```
     Survived  Pclass                                               Name  \
0           0       3                            Braund, Mr. Owen Harris
1           1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
2           1       3                             Heikkinen, Miss. Laina
3           1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
4           0       3                           Allen, Mr. William Henry
..        …     …                                                    …
886         0       2                              Montvila, Rev. Juozas
887         1       1                       Graham, Miss. Margaret Edith
888         0       3           Johnston, Miss. Catherine Helen "Carrie"
889         1       1                              Behr, Mr. Karl Howell
890         0       3                                Dooley, Mr. Patrick

        Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
0      male  22.0      1      0         A/5 21171   7.2500   NaN        S
1    female  38.0      1      0          PC 17599  71.2833   C85        C
2    female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
3    female  35.0      1      0            113803  53.1000  C123        S
4      male  35.0      0      0            373450   8.0500   NaN        S
..      …   …    …    …               …       …   …      …
886    male  27.0      0      0            211536  13.0000   NaN        S
887  female  19.0      0      0            112053  30.0000   B42        S
888  female   NaN      1      2        W./C. 6607  23.4500   NaN        S
889    male  26.0      0      0            111369  30.0000  C148        C
890    male  32.0      0      0            370376   7.7500   NaN        Q

[891 rows x 11 columns]
```

1.1 Exploratory Data Analysis (EDA)

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

[5]:
```
     Survived  Pclass                                          Name     Sex   Age  \
886         0       2                         Montvila, Rev. Juozas    male  27.0
887         1       1                  Graham, Miss. Margaret Edith  female  19.0
888         0       3  Johnston, Miss. Catherine Helen "Carrie"  female   NaN
889         1       1                         Behr, Mr. Karl Howell    male  26.0
890         0       3                           Dooley, Mr. Patrick    male  32.0
```

	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	0	0	211536	13.00	NaN	S
887	0	0	112053	30.00	B42	S
888	1	2	W./C. 6607	23.45	NaN	S
889	0	0	111369	30.00	C148	C
890	0	0	370376	7.75	NaN	Q

[6]:
```
len(df)
```

[6]: 891

[7]:
```
df.shape
```

[7]: (891, 11)

[8]:
```
df.describe()
```

[8]:

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Conclusiones:

- Existen columnas de “missing values” (Valores que faltan)

[9]:
```
# y aqui vemos cuantas columnas tiene valores que faltan
df.isnull().sum()
```

[9]:
```
Survived      0
Pclass        0
Name          0
Sex           0
Age         177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

[10]:
```
df.Cabin.value_counts()
```

[10]:
```
Cabin
G6             4
C23 C25 C27    4
B96 B98        4
F2             3
D              3
               ..
E17            1
A24            1
C50            1
B42            1
C148           1
Name: count, Length: 147, dtype: int64
```

[11]:
```
for cabina in df.Cabin:
    print(cabina)
```

```
nan
C85
nan
C123
nan
nan
E46
nan
nan
nan
G6
C103
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
D56
nan
A6
nan
nan
nan
C23 C25 C27
nan
nan
nan
```

B78
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
D33
nan
B30
C52
nan
nan
nan
nan
nan
B28
C83
nan
nan
nan
F33
nan
nan
nan
nan
nan
nan
nan
nan
F G73
nan
nan
nan

```
nan
nan
nan
nan
nan
nan
nan
nan
nan
C23 C25 C27
nan
nan
nan
E31
nan
nan
nan
A5
D10 D12
nan
nan
nan
nan
D26
nan
nan
nan
nan
nan
nan
nan
C110
nan
nan
nan
nan
nan
nan
nan
B58 B60
nan
nan
nan
nan
E101
D26
nan
nan
```

nan
F E69
nan
nan
nan
nan
nan
nan
nan
D47
C123
nan
B86
nan
nan
nan
nan
nan
nan
nan
nan
nan
F2
nan
nan
C2
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
E33
nan
nan
nan
B19
nan
nan
nan
A7

nan
nan
C49
nan
nan
nan
nan
nan
F4
nan
A32
nan
nan
nan
nan
nan
nan
nan
F2
B4
B80
nan
nan
nan
nan
nan
nan
nan
nan
nan
G6
nan
nan
nan
A31
nan
nan
nan
nan
nan
D36
nan
nan
D15
nan
nan
nan
nan

nan
C93
nan
nan
nan
nan
nan
C83
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
C78
nan
nan
D35
nan
nan
G6
C87
nan
nan
nan
nan
B77
nan
nan
nan
nan
E67
B94
nan
nan
nan
nan
C125
C99
nan

```
nan
nan
C118
nan
D7
nan
nan
nan
nan
nan
nan
nan
nan
nan
A19
nan
nan
nan
nan
nan
nan
nan
B49
D
nan
nan
nan
nan
C22 C26
C106
B58 B60
nan
nan
nan
E101
nan
C22 C26
nan
C65
nan
E36
C54
B57 B59 B63 B66
nan
nan
nan
nan
nan
nan
C7
```

E34
nan
nan
nan
nan
nan
C32
nan
D
nan
B18
nan
C124
C91
nan
nan
nan
C2
E40
nan
T
F2
C23 C25 C27
nan
nan
nan
F33
nan
nan
nan
nan
nan
C128
nan
nan
nan
nan
E33
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
D37

nan
nan
B35
E50
nan
nan
nan
nan
nan
nan
nan
C82
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
B96 B98
nan
nan
D36
G6
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
C78
nan
nan

```
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
E10
C52
nan
nan
nan
E44
B96 B98
nan
nan
C23 C25 C27
nan
nan
nan
nan
nan
nan
A34
nan
nan
nan
C104
nan
nan
C111
C92
nan
nan
E38
D21
nan
nan
E12
nan
E63
```

nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
D
nan
A14
nan
nan
nan
nan
nan
nan
nan
nan
nan
B49
nan
C93
B37
nan
nan
nan
nan
C30
nan
nan
nan
D20
nan
C22 C26
nan
nan
nan
nan
nan
B79
C65
nan
nan
nan
nan
nan

nan
E25
nan
nan
D46
F33
nan
nan
nan
B73
nan
nan
B18
nan
nan
nan
C95
nan
nan
nan
nan
nan
nan
nan
nan
B38
nan
nan
B39
B22
nan
nan
nan
C86
nan
nan
nan
nan
nan
C70
nan
nan
nan
nan
nan
A16
nan
E67

nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
nan
C101
E25
nan
nan
nan
nan
E44
nan
nan
nan
C68
nan
A10
nan
E68
nan
B41
nan
nan
nan
D20
nan
nan
nan
nan
nan
nan
nan
A20
nan
nan
nan
nan
nan
nan
nan

nan
nan
C125
nan
nan
nan
nan
nan
nan
nan
nan
F4
nan
nan
D19
nan
nan
nan
D50
nan
D9
nan
nan
A23
nan
B50
nan
nan
nan
nan
nan
nan
nan
nan
B35
nan
nan
nan
D33
nan
A26
nan
nan
nan
nan
nan
nan
nan

```
nan
nan
nan
nan
D48
nan
nan
E58
nan
nan
nan
nan
nan
nan
nan
C126
nan
B71
nan
nan
nan
nan
nan
nan
nan
nan
B51 B53 B55
nan
D49
nan
nan
nan
nan
nan
nan
nan
nan
B5
B20
nan
nan
nan
nan
nan
nan
nan
nan
C68
F G63
C62 C64
E24
nan
```

```
nan
nan
nan
nan
E24
nan
nan
C90
C124
C126
nan
nan
F G73
C45
E101
nan
nan
nan
nan
nan
nan
E8
nan
nan
nan
nan
nan
B5
nan
nan
nan
nan
nan
nan
B101
nan
nan
D45
C46
B57 B59 B63 B66
nan
nan
B22
nan
nan
D30
nan
nan
```

E121
nan
nan
nan
nan
nan
nan
nan
B77
nan
nan
nan
B96 B98
nan
D11
nan
nan
nan
nan
nan
nan
nan
E77
nan
nan
nan
F38
nan
nan
B3
nan
B20
D6
nan
nan
nan
nan
nan
nan
B82 B84
nan
nan
nan
nan
nan
nan
D17
nan
nan

```
nan
nan
nan
B96 B98
nan
nan
nan
A36
nan
nan
E8
nan
nan
nan
nan
nan
B102
nan
nan
nan
nan
B69
nan
nan
E121
nan
nan
nan
nan
nan
B28
nan
nan
nan
nan
nan
E49
nan
nan
nan
C47
nan
nan
nan
nan
nan
nan
nan
nan
```

```
nan
nan
C92
nan
nan
nan
D28
nan
nan
nan
E17
nan
nan
nan
nan
D17
nan
nan
nan
nan
A24
nan
nan
nan
D35
B51 B53 B55
nan
nan
nan
nan
nan
nan
C50
nan
nan
nan
nan
nan
nan
nan
nan
B42
nan
C148
nan
```

[12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Name      891 non-null    object
 3   Sex       891 non-null    object
 4   Age       714 non-null    float64
 5   SibSp     891 non-null    int64
 6   Parch     891 non-null    int64
 7   Ticket    891 non-null    object
 8   Fare      891 non-null    float64
 9   Cabin     204 non-null    object
 10  Embarked  889 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 76.7+ KB
```

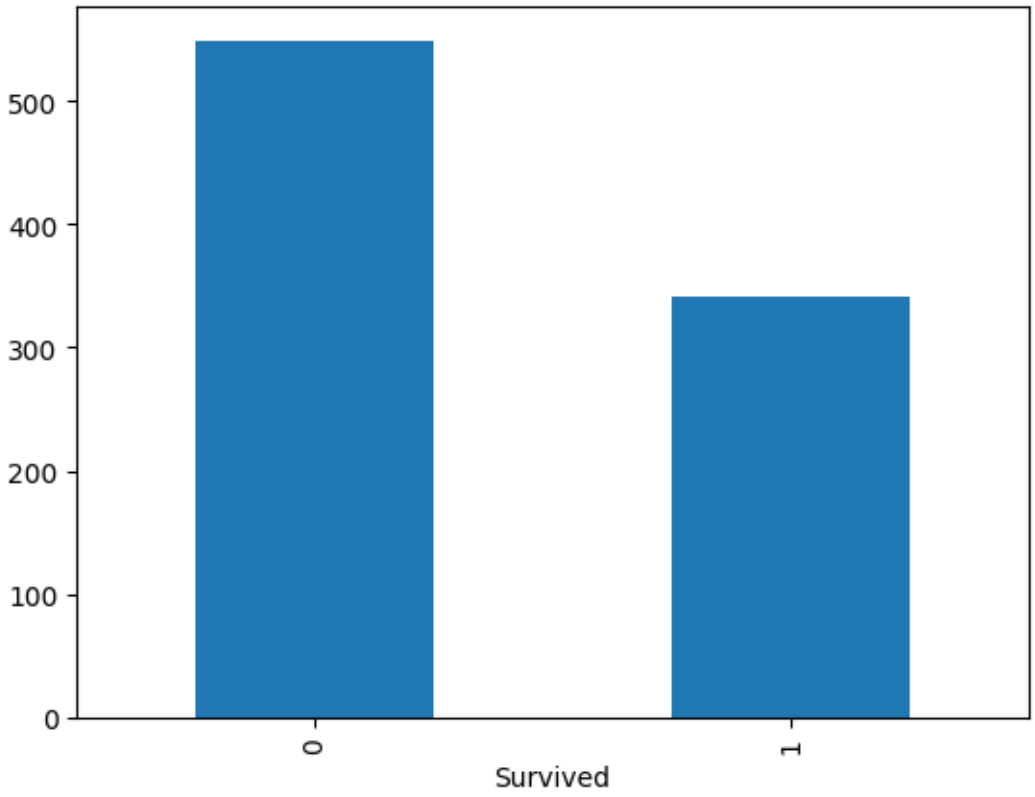
[13]:
```
df.Survived.value_counts()
```

[13]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

[14]:
```
df.Survived.value_counts(normalize=True)
```

[14]:
```
Survived
0    0.616162
1    0.383838
Name: proportion, dtype: float64
```

[15]:
```
df.Survived.value_counts().plot(kind="bar")
plt.show()
```



1.2 ¿Cómo seleccionar información concreta de nuestro dataset?

Forma 1

[16]:
```
df["Age"].head()
```

[16]:
```
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
Name: Age, dtype: float64
```

Forma 2

[17]:
```
df.Age.head()
```

[17]:
```
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
```

```
Name: Age, dtype: float64
```

Forma 3

[18]:
```
df[["Age"]].head()
```

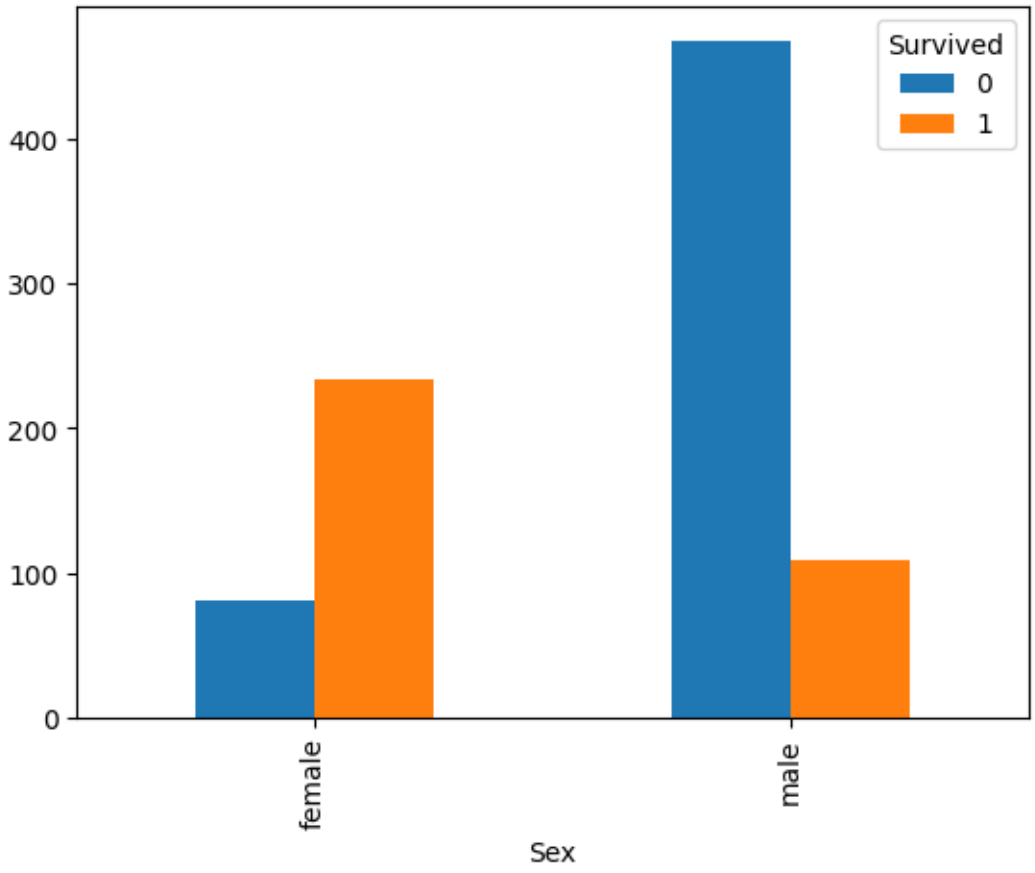
[18]:
```
    Age
0  22.0
1  38.0
2  26.0
3  35.0
4  35.0
```

1.3 Crosstab

[19]:
```
pd.crosstab(df.Sex, df.Survived)
```

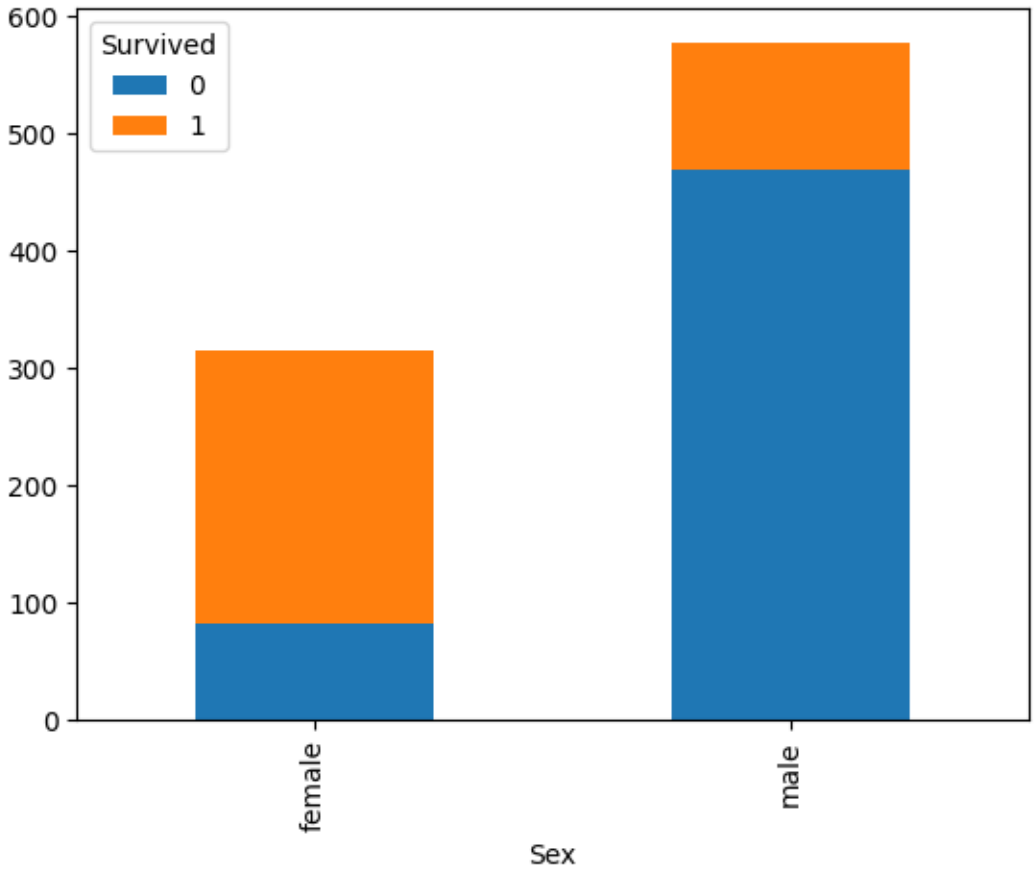
[19]:
```
Survived    0    1
Sex
female     81  233
male      468  109
```

[20]:
```
pd.crosstab(df.Sex, df.Survived).plot(kind="bar")
plt.show()
```



Conclusión: * La mayoría de las mujeres sobreviven. * La mayoría de los hombres NO sobrevivieron

[21]:
```
pd.crosstab(df.Sex, df.Survived).plot(kind="bar", stacked=True)
plt.show()
```

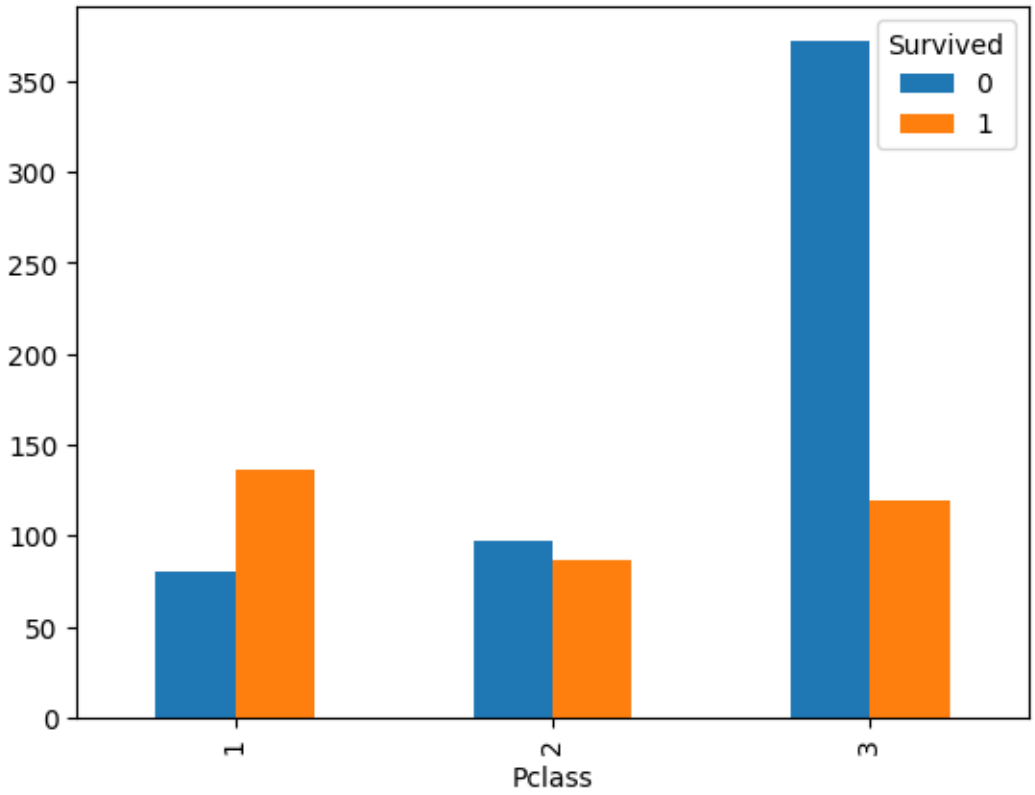


Conclusión: * Hay más hombres que mujeres, es casi el doble.

[22]:
```
pd.crosstab(df.Pclass, df.Survived)
```

[22]:
```
Survived    0    1
Pclass
1          80  136
2          97   87
3         372  119
```

[23]:
```
pd.crosstab(df.Pclass, df.Survived).plot(kind="bar")
plt.show()
```



Conclusión:

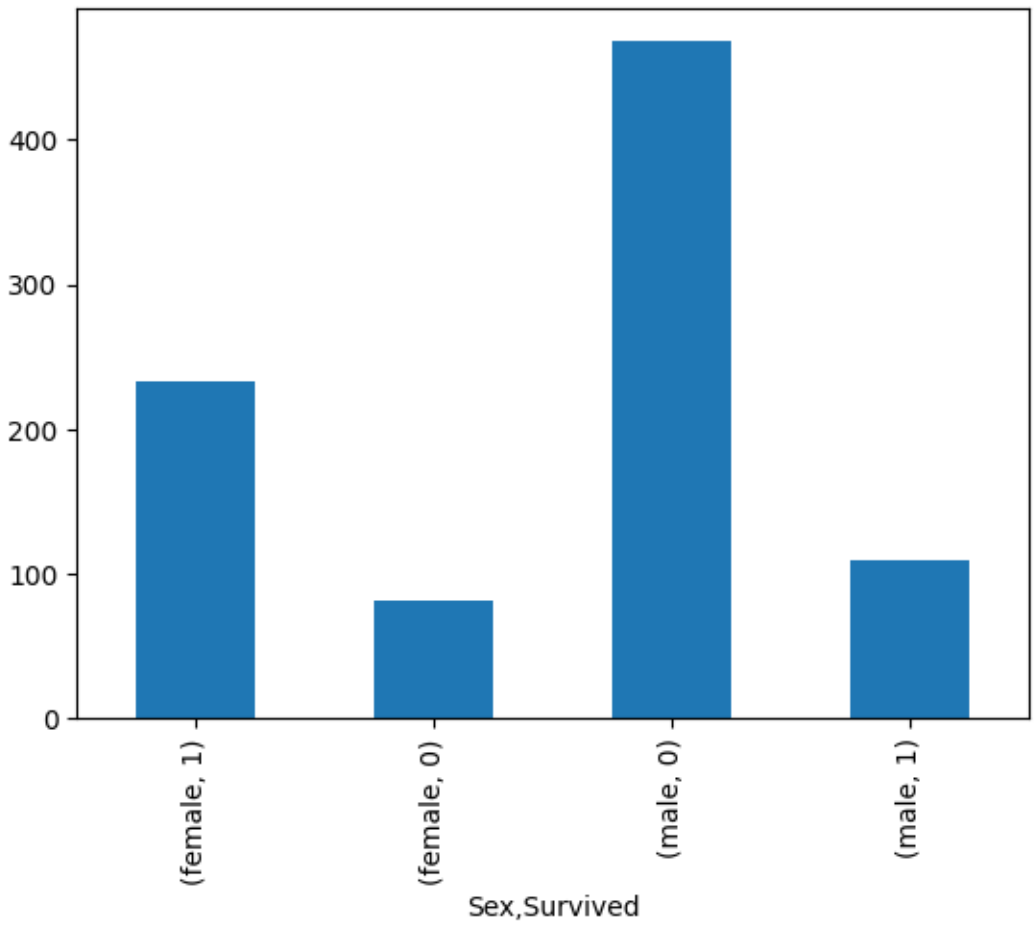
- La mayoría de los que NO sobrevivieron eran de la 3ª clase

1.4 groupby

[24]:
```
df.groupby("Sex").Survived.value_counts()
```

[24]:
```
Sex     Survived
female  1           233
        0            81
male    0           468
        1           109
Name: count, dtype: int64
```

[25]:
```
df.groupby("Sex").Survived.value_counts().plot(kind="bar")
plt.show()
```



1.5 por filtrado

- Selecciono aquellas filas donde Pclass == 1
- Me creo un dataframe de la misma forma que tenía antes

[26]:
```
# Una forma...
```

[27]:
```
df_sex_uno = df[df.Pclass == 1]
df_sex_uno.head()
```

[27]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…								
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)								
6	0	1	McCarthy, Mr. Timothy J								
11	1	1	Bonnell, Miss. Elizabeth								
23	1	1	Sloper, Mr. William Thompson								

```
1   female  38.0      1      0  PC 17599  71.2833   C85        C
3   female  35.0      1      0    113803  53.1000  C123        S
6     male  54.0      0      0     17463  51.8625   E46        S
11  female  58.0      0      0    113783  26.5500  C103        S
23    male  28.0      0      0    113788  35.5000    A6        S
```

[28]:
```
# Otra forma...
```

[29]:
```
df_sex_crosstab = df[df.Pclass == 1]["Survived"]
df_sex_crosstab.head()
```

[29]:
```
1     1
3     1
6     0
11    1
23    1
Name: Survived, dtype: int64
```

1.6 Ejemplos de creación de dataframes

[30]:
```
df_sobreviven_todos = df[df["Survived"] == 1]
df_sobreviven_ninguno = df[df["Survived"] == 0]
hombres_sobrevivieron = df[(df["Survived"] == 1) & (df["Sex"] == "male")]
hombres__no_sobrevivieron = df[(df["Survived"] == 0) & (df["Sex"] == "male")]
mujeres_sobrevivieron = df[(df["Survived"] == 1) & (df["Sex"] == "female")]
mujeres_no_sobrevivieron = df[(df["Survived"] == 0) & (df["Sex"] == "female")]
```

[31]:
```
df_sobreviven_todos.head()
```

[31]:
```
   Survived  Pclass                                               Name  \
1         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
2         1       3                             Heikkinen, Miss. Laina
3         1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
8         1       3  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
9         1       2                Nasser, Mrs. Nicholas (Adele Achem)

      Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
1  female  38.0      1      0          PC 17599  71.2833   C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
3  female  35.0      1      0            113803  53.1000  C123        S
8  female  27.0      0      2            347742  11.1333   NaN        S
9  female  14.0      1      0            237736  30.0708   NaN        C
```

[32]:
```
df_sobreviven_todos.Survived.value_counts(3)
```

[32]:
```
Survived
1    1.0
```

```
Name: proportion, dtype: float64
```

[33]:
```
df_sobreviven_ninguno.head()
```

[33]:
```
   Survived  Pclass                                Name   Sex   Age  SibSp  Parch  \
0         0       3             Braund, Mr. Owen Harris  male  22.0      1      0
4         0       3            Allen, Mr. William Henry  male  35.0      0      0
5         0       3                    Moran, Mr. James  male   NaN      0      0
6         0       1             McCarthy, Mr. Timothy J  male  54.0      0      0
7         0       3      Palsson, Master. Gosta Leonard  male   2.0      3      1

      Ticket     Fare Cabin Embarked
0  A/5 21171   7.2500   NaN        S
4     373450   8.0500   NaN        S
5     330877   8.4583   NaN        Q
6      17463  51.8625   E46        S
7     349909  21.0750   NaN        S
```

[34]:
```
df_sobreviven_ninguno.Survived.value_counts(3)
```

[34]:
```
Survived
0    1.0
Name: proportion, dtype: float64
```

[35]:
```
hombres_sobrevivieron.head()
```

[35]:
```
    Survived  Pclass                              Name   Sex   Age  SibSp  Parch  \
17         1       2      Williams, Mr. Charles Eugene  male   NaN      0      0
21         1       2             Beesley, Mr. Lawrence  male  34.0      0      0
23         1       1      Sloper, Mr. William Thompson  male  28.0      0      0
36         1       3                   Mamee, Mr. Hanna  male   NaN      0      0
55         1       1                  Woolner, Mr. Hugh  male   NaN      0      0

    Ticket     Fare Cabin Embarked
17  244373  13.0000   NaN        S
21  248698  13.0000   D56        S
23  113788  35.5000    A6        S
36    2677   7.2292   NaN        C
55   19947  35.5000   C52        S
```

[36]:
```
hombres__no_sobrevivieron.head()
```

[36]:
```
   Survived  Pclass                                Name   Sex   Age  SibSp  Parch  \
0         0       3             Braund, Mr. Owen Harris  male  22.0      1      0
4         0       3            Allen, Mr. William Henry  male  35.0      0      0
5         0       3                    Moran, Mr. James  male   NaN      0      0
6         0       1             McCarthy, Mr. Timothy J  male  54.0      0      0
7         0       3      Palsson, Master. Gosta Leonard  male   2.0      3      1
```

```
      Ticket     Fare Cabin Embarked
0  A/5 21171   7.2500   NaN        S
4     373450   8.0500   NaN        S
5     330877   8.4583   NaN        Q
6      17463  51.8625   E46        S
7     349909  21.0750   NaN        S
```

[37]:
```
mujeres_sobrevivieron.head()
```

[37]:
```
   Survived  Pclass                                                Name  \
1         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
2         1       3                             Heikkinen, Miss. Laina
3         1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
8         1       3  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
9         1       2                Nasser, Mrs. Nicholas (Adele Achem)

      Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
1  female  38.0      1      0          PC 17599  71.2833   C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
3  female  35.0      1      0            113803  53.1000  C123        S
8  female  27.0      0      2            347742  11.1333   NaN        S
9  female  14.0      1      0            237736  30.0708   NaN        C
```

[38]:
```
mujeres_no_sobrevivieron.head()
```

[38]:
```
    Survived  Pclass                                               Name  \
14         0       3               Vestrom, Miss. Hulda Amanda Adolfina
18         0       3  Vander Planke, Mrs. Julius (Emelia Maria Vande…
24         0       3                      Palsson, Miss. Torborg Danira
38         0       3                 Vander Planke, Miss. Augusta Maria
40         0       3     Ahlin, Mrs. Johan (Johanna Persdotter Larsson)

       Sex   Age  SibSp  Parch  Ticket     Fare Cabin Embarked
14  female  14.0      0      0  350406   7.8542   NaN        S
18  female  31.0      1      0  345763  18.0000   NaN        S
24  female   8.0      3      1  349909  21.0750   NaN        S
38  female  18.0      2      0  345764  18.0000   NaN        S
40  female  40.0      1      0    7546   9.4750   NaN        S
```

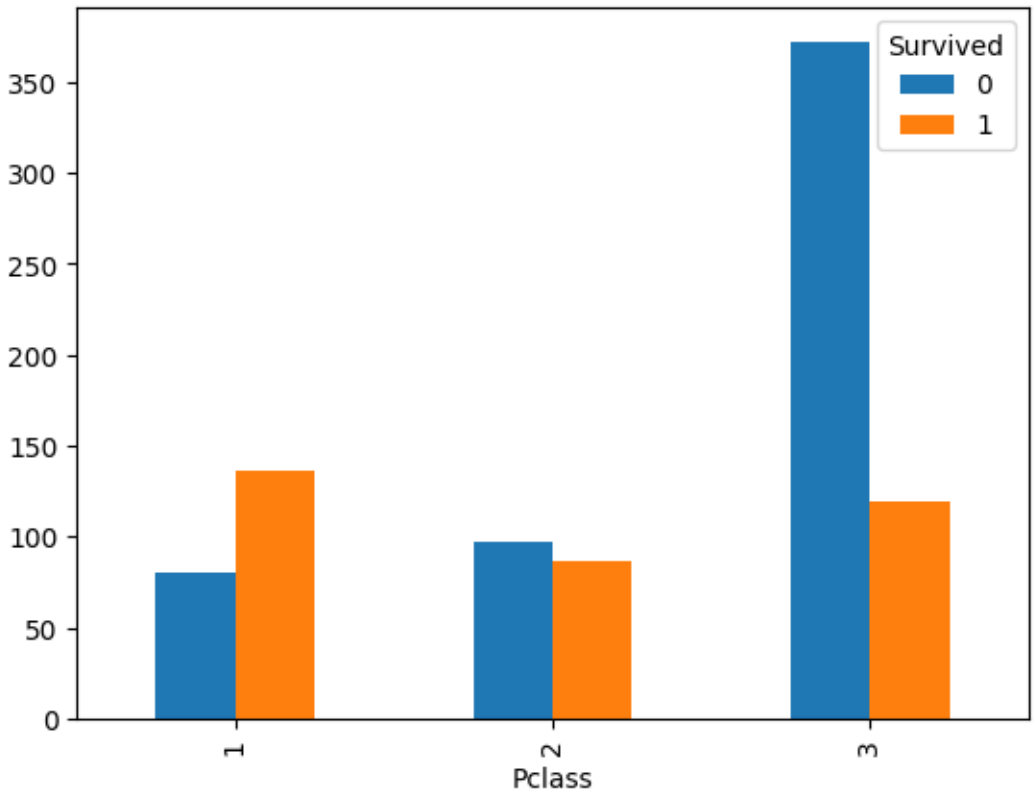
2 Obtenemos información de los gráficos

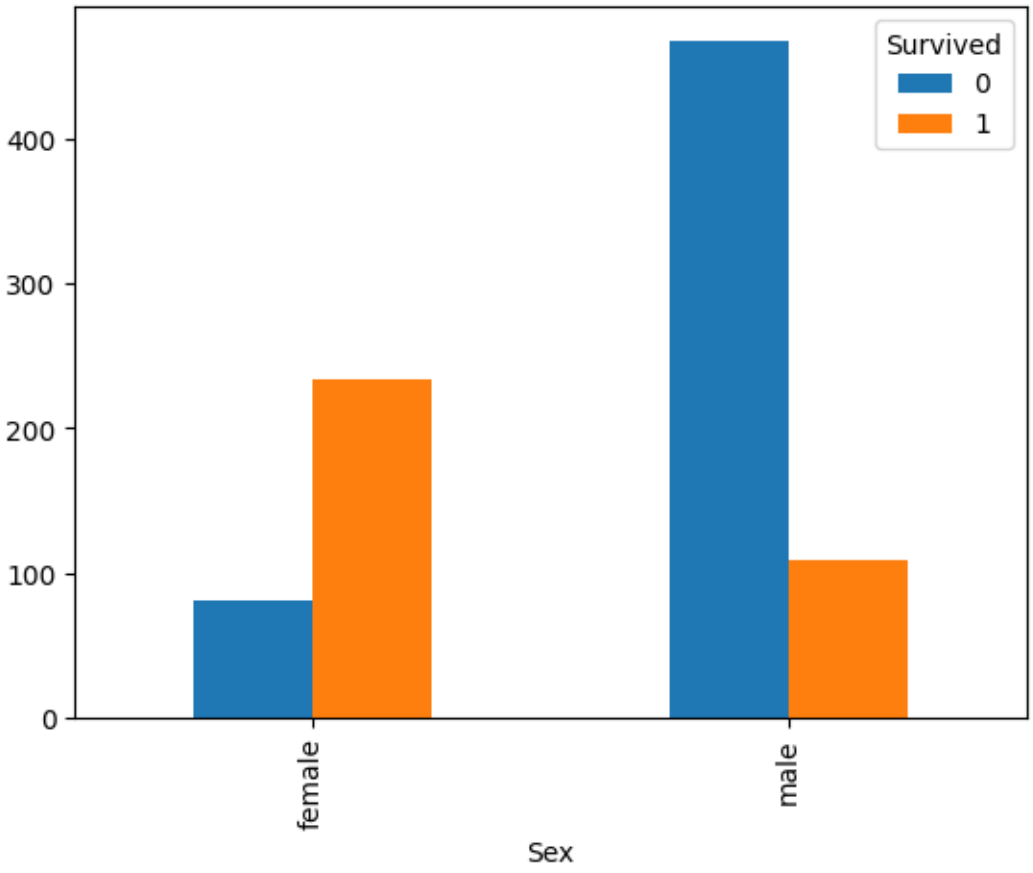
[39]:
```
df.head()
```

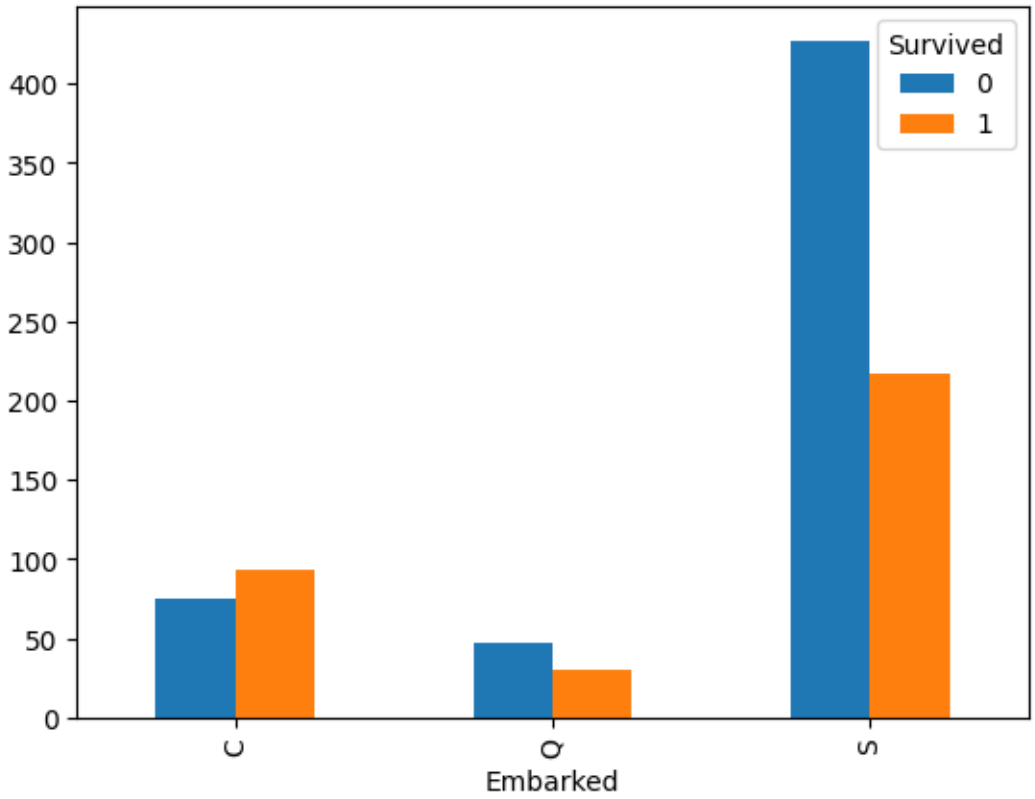
[39]:
```
   Survived  Pclass                                                Name  \
0         0       3                            Braund, Mr. Owen Harris
1         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
```

```
2         1       3                             Heikkinen, Miss. Laina
3         1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
4         0       3                           Allen, Mr. William Henry

      Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
0    male  22.0      1      0         A/5 21171   7.2500   NaN        S
1  female  38.0      1      0          PC 17599  71.2833   C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
3  female  35.0      1      0            113803  53.1000  C123        S
4    male  35.0      0      0            373450   8.0500   NaN        S
```

[40]:
```
opciones = ["Pclass", "Sex", "Embarked"]

for opcion in opciones:
    pd.crosstab(df[opcion], df.Survived).plot(kind="bar")
    plt.show()
```

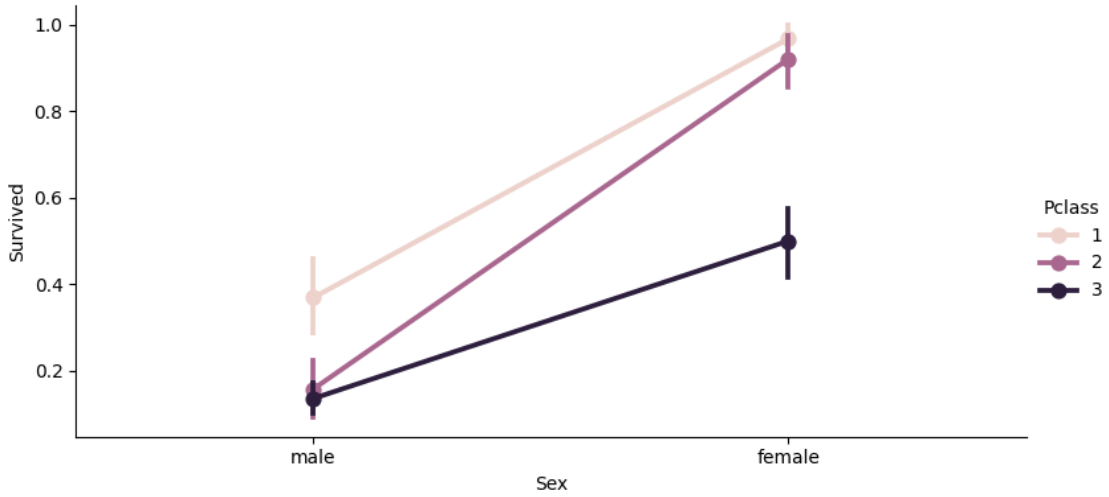



Survived
0
1
400
300
200
100
0
female
male
Sex

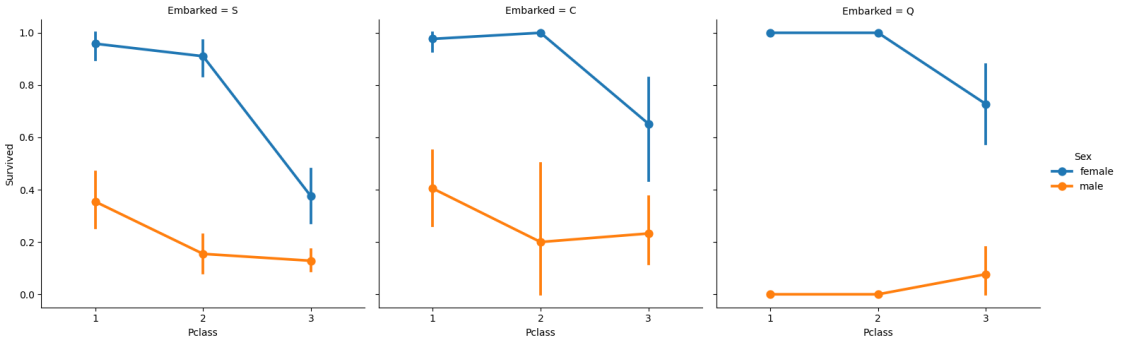


SEABORN

[41]:
```
sns.catplot(x="Sex", y="Survived", hue="Pclass", kind="point", height=4,␣
↪aspect=2, data=df)
plt.show()
```



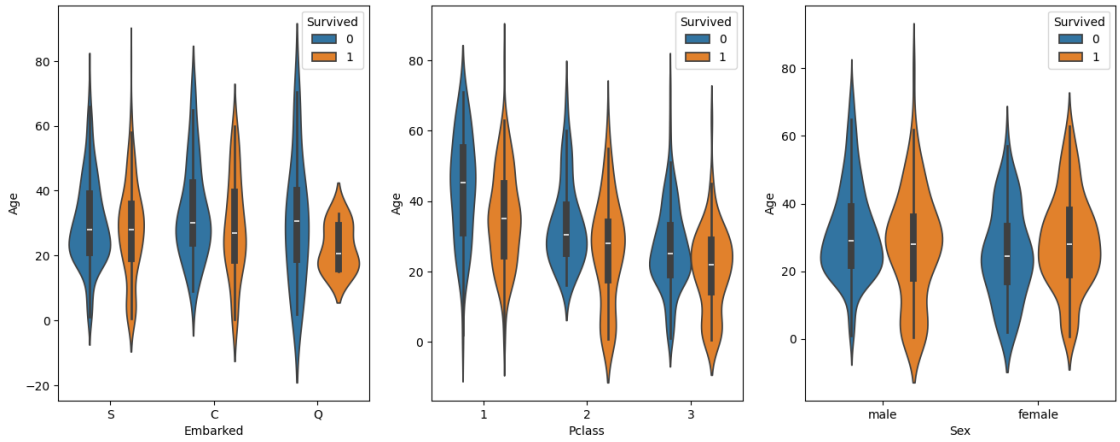
[42]:
```
sns.catplot(x="Pclass", y="Survived", hue="Sex", kind="point", col="Embarked",␣
↪data=df)
plt.show()
```



Algunas conclusiones: * Nos fijamos en la gráfica de la izquierda, embarked=“S” –> las mujeres de 3 clase que embarcaron en “S” fallecieron muchas en comparación con 1 y 2 clase, pese a ello sobrevivieron algo más que los hombres de 1 clase del mismo puerto. * los hombres con mayor porcentaje e supervivencia embarcaron en “C” * Los hombres con menor porcentaje de supervivencia embarcaron en “Q” * Vemos nuevamente como la mayoría de las mujeres sobrevivieron, pero no los hombres.

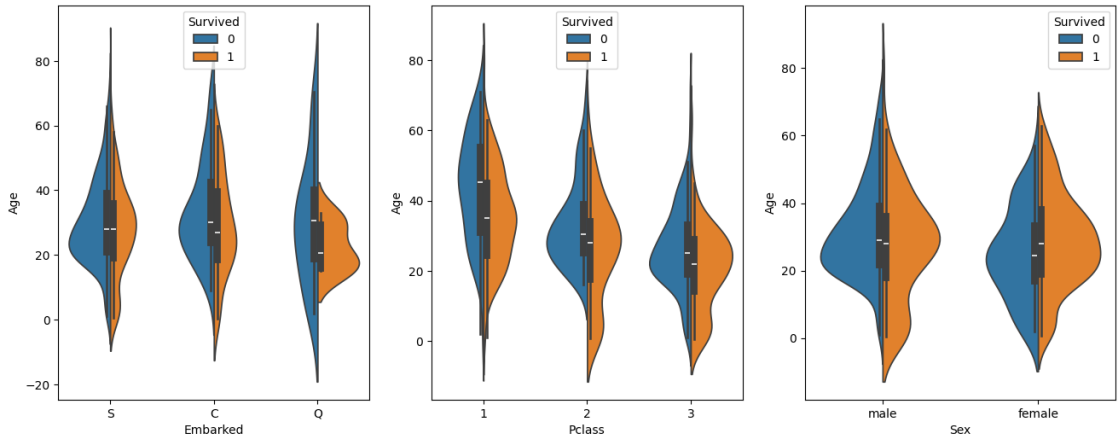
3 Edad y supervivencia

[43]:
```
# me creo una figura
fig = plt.figure(figsize=(16,6))
# 3 subplots
# 1 fila 3 columnas - gráfica 1
ax1 = fig.add_subplot(131)
# 1 fila 3 columnas - gráfica 2
ax2 = fig.add_subplot(132)
# 1 fila 3 columnas - gráfica 3
ax3 = fig.add_subplot(133)

# violinplot
sns.violinplot(x="Embarked", y="Age", hue="Survived", data=df, ax=ax1)
sns.violinplot(x="Pclass", y="Age", hue="Survived", data=df, ax=ax2)
sns.violinplot(x="Sex", y="Age", hue="Survived", data=df, ax=ax3)

plt.show()
```



Hago un split== True, para mostrarlo más visual

[44]:
```
# me creo una figura
fig = plt.figure(figsize=(16,6))
# 3 subplots
# 1 fila 3 columnas - gráfica 1
ax1 = fig.add_subplot(131)
# 1 fila 3 columnas - gráfica 2
ax2 = fig.add_subplot(132)
# 1 fila 3 columnas - gráfica 3
ax3 = fig.add_subplot(133)

# violinplot
sns.violinplot(x="Embarked", y="Age", hue="Survived", data=df, split=True,␣
 ↪ax=ax1)
sns.violinplot(x="Pclass", y="Age", hue="Survived", data=df, split=True, ax=ax2)
sns.violinplot(x="Sex", y="Age", hue="Survived", data=df, split=True, ax=ax3)

plt.show()
```



Conclusiones:

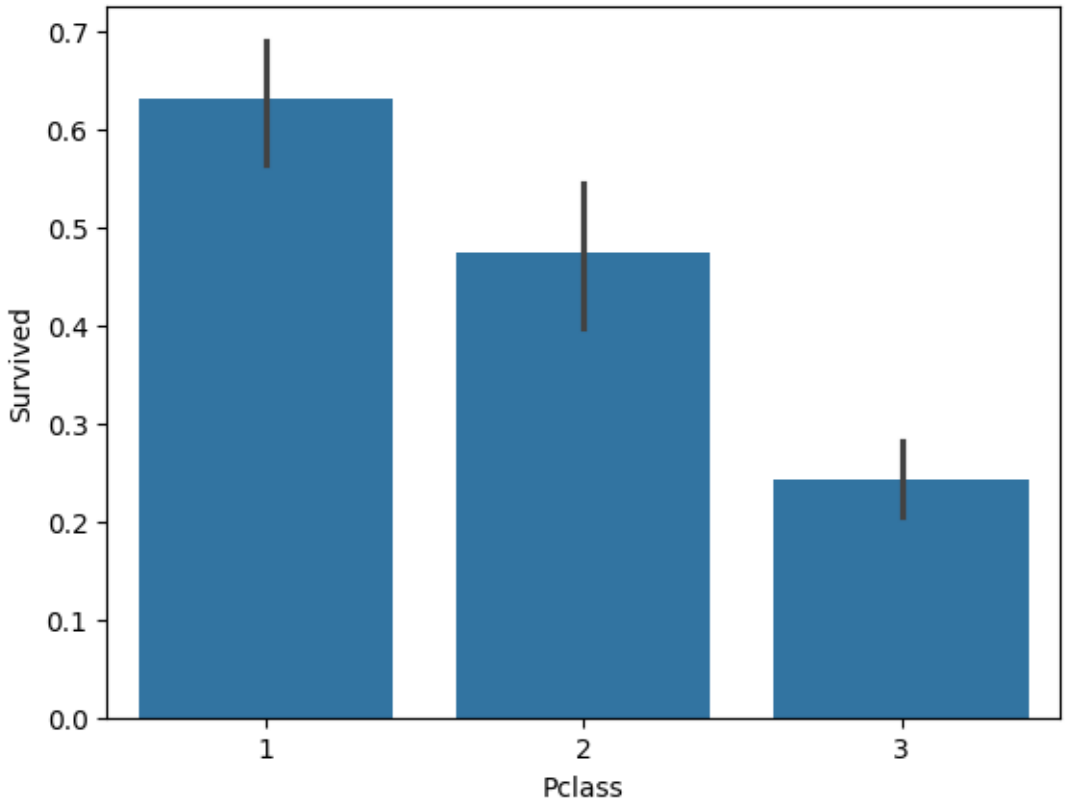
- EMBARKED y Age:
  - La gente de unos 18-35 años de Q SI sobrevivieron mayoritariamente,(no todos)
  - no hay porcentajes mayoritarios significativos en las otras 2 embarcaciones
  - En Q embarcaron bastantes niños los cuales no sobrevivieron.
- PCLASS y Age:
  - De la 2ª clase sobre todo y la 3 sobrevivieron la mayoria de sus niños
- Sex y Age:
  - Hay mas ancianos que ancianas
  - Los jovenes (varón) menores de 20 años en general sobrevivieron pero no las mujeres

[45]:
```
df.Age.describe()
```

[45]:
```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

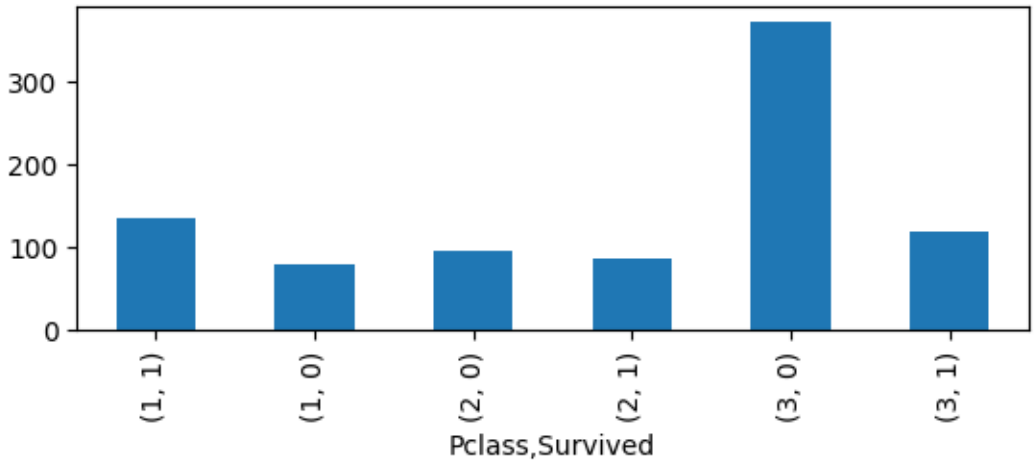
3.0.1 barplot

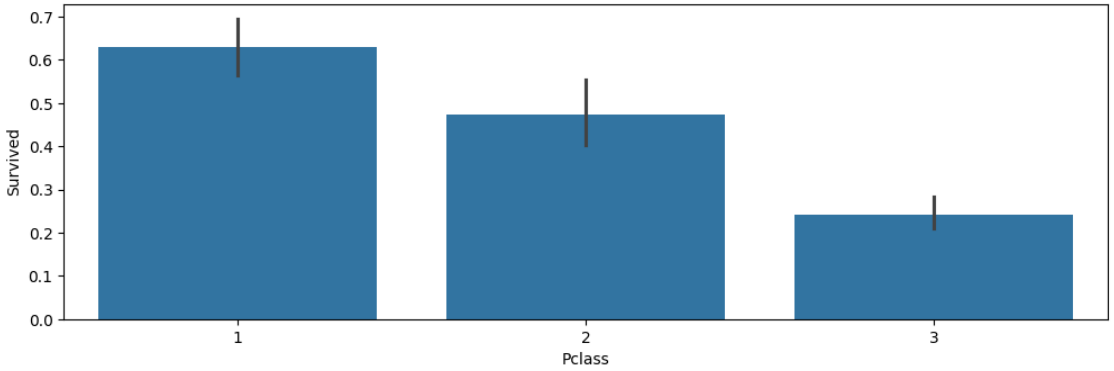
[46]:
```
sns.barplot(x="Pclass", y="Survived", data=df)
plt.show()
```



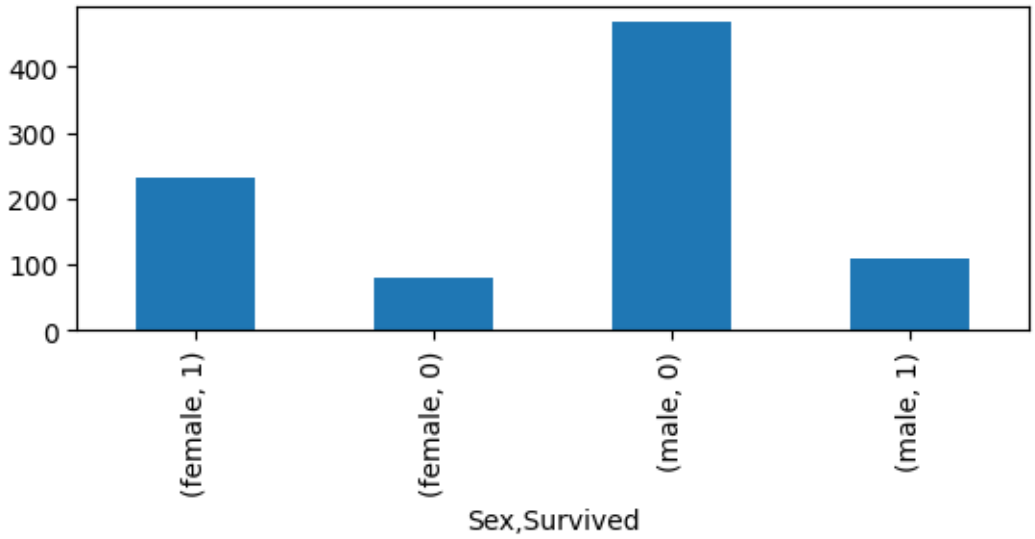
[47]:
```
def funcion_graficas(feat):
    plt.subplot(2, 1, 1)
    df.groupby(feat).Survived.value_counts().plot(kind="bar")
    plt.figure(figsize=(12,8))
    plt.subplot(2, 1, 2)
    sns.barplot(x=feat, y="Survived", data=df)
    plt.show()
```

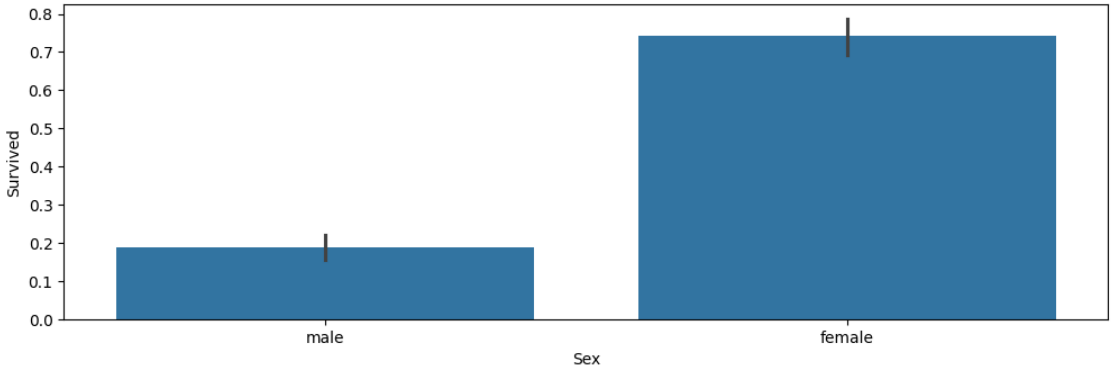
[48]:
```
funcion_graficas("Pclass")
```



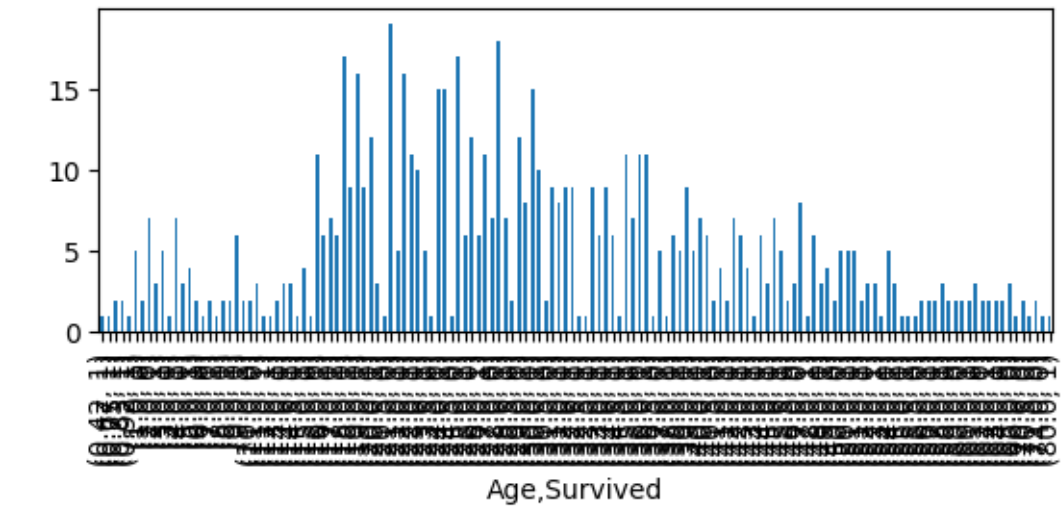


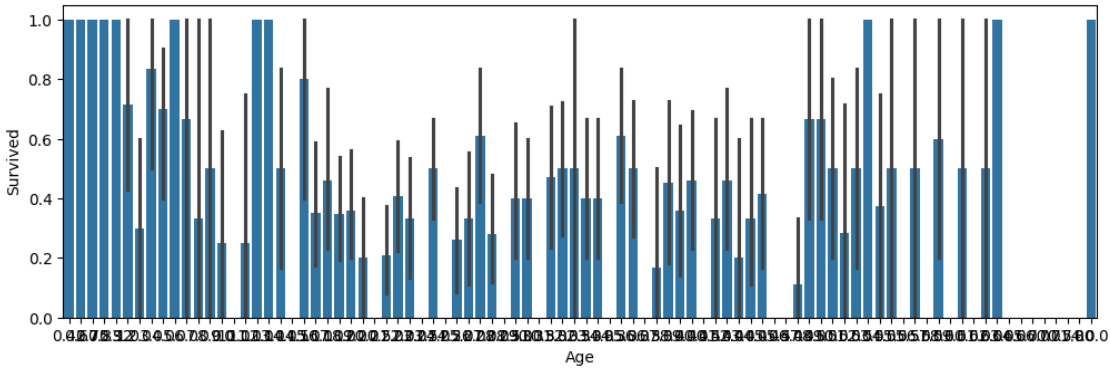
[49]:
```
funcion_graficas("Sex")
```



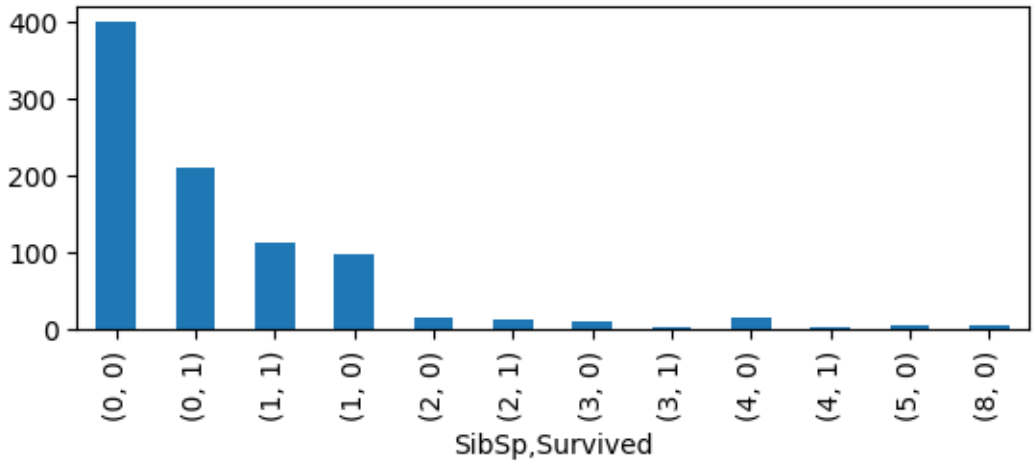


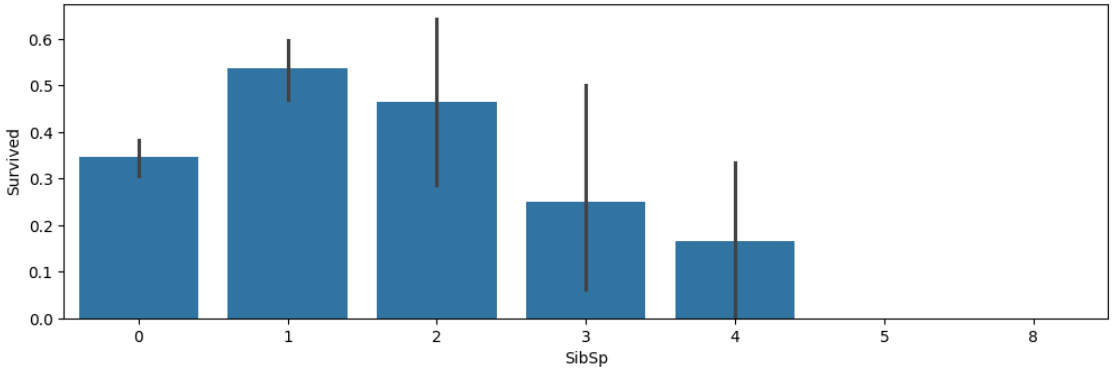
[50]:
```
funcion_graficas("Age")
```



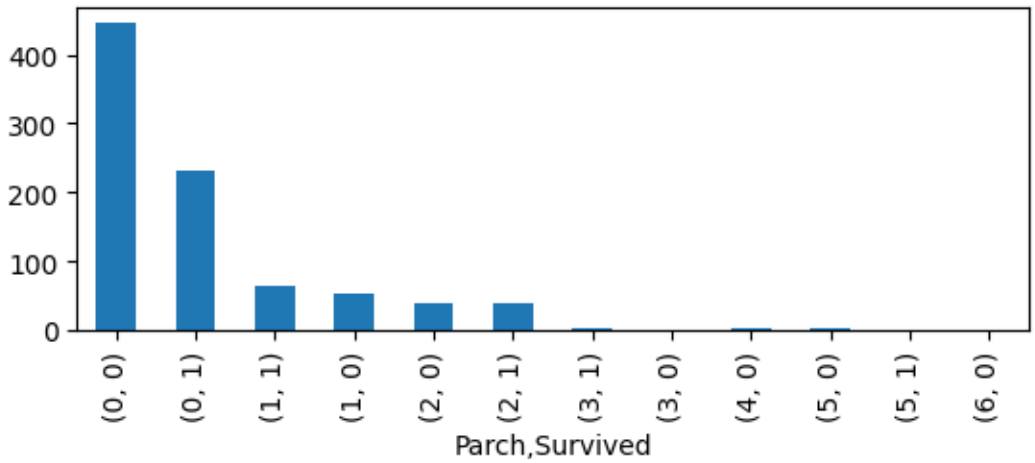


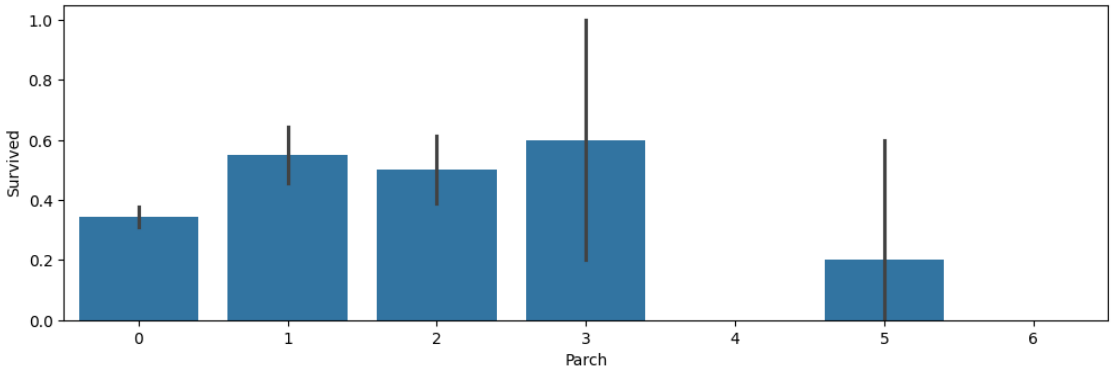
[51]:
```
funcion_graficas("SibSp")
```



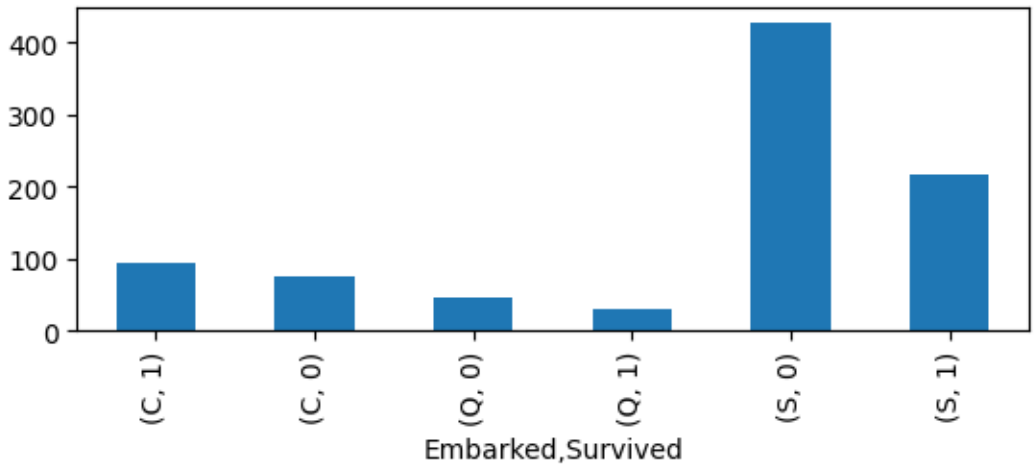


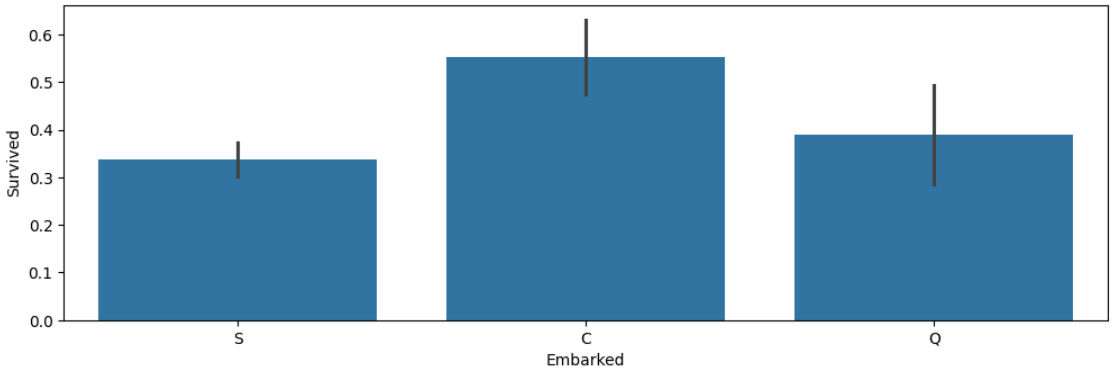
[52]:
```
funcion_graficas("Parch")
```





[53]:
```
funcion_graficas("Embarked")
```





3.1 Feature Engineering

En esta parte podemos hacer uso de la información obtenida y conclusiones.

Para hacerlo lo más simple posible, lo que haremos será elegir solamente algunas columnas.

[54]:
```
df.head()
```

[54]:

	Survived	Pclass	Name
0	0	3	Braund, Mr. Owen Harris
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…
2	1	3	Heikkinen, Miss. Laina
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)
4	0	3	Allen, Mr. William Henry

```
      Sex   Age  SibSp  Parch            Ticket     Fare Cabin Embarked
0    male  22.0      1      0         A/5 21171   7.2500   NaN        S
1  female  38.0      1      0          PC 17599  71.2833   C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250   NaN        S
3  female  35.0      1      0            113803  53.1000  C123        S
4    male  35.0      0      0            373450   8.0500   NaN        S
```

[55]:
```
df.isnull().sum()
```

[55]:
```
Survived      0
Pclass        0
Name          0
Sex           0
Age         177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

-1-Name- no lo tendremos en cuenta por simplificar

df[“Name”] = df[“Name”].str.extract(“([A-Za-z]+)”, expand=False)

seria posible una posible forma de analizar la columna Name, pero no lo haremos.

-2-Age- Usamos el valor promedio de la columna para rellernar los valores que faltan

[56]:
```
df.Age.isnull().sum()
```

[56]:
```
np.int64(177)
```

[57]:
```
df.Age = df.Age.fillna(df.Age.mean())
```

[58]:
```
df.Age.isnull().sum()
```

[58]:
```
np.int64(0)
```

[59]:
```
df
```

[59]:
```
     Survived  Pclass                                               Name  \
0           0       3                            Braund, Mr. Owen Harris
1           1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
2           1       3                             Heikkinen, Miss. Laina
3           1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
4           0       3                           Allen, Mr. William Henry
..        …     …                                                    …
```

```
886         0       2                                Montvila, Rev. Juozas
887         1       1                         Graham, Miss. Margaret Edith
888         0       3             Johnston, Miss. Catherine Helen "Carrie"
889         1       1                                Behr, Mr. Karl Howell
890         0       3                                  Dooley, Mr. Patrick

        Sex        Age  SibSp  Parch            Ticket     Fare Cabin Embarked
0      male  22.000000      1      0         A/5 21171   7.2500   NaN        S
1    female  38.000000      1      0          PC 17599  71.2833   C85        C
2    female  26.000000      0      0  STON/O2. 3101282   7.9250   NaN        S
3    female  35.000000      1      0            113803  53.1000  C123        S
4      male  35.000000      0      0            373450   8.0500   NaN        S
..      …        …    …    …                …       …   …       …
886    male  27.000000      0      0            211536  13.0000   NaN        S
887  female  19.000000      0      0            112053  30.0000   B42        S
888  female  29.699118      1      2        W./C. 6607  23.4500   NaN        S
889    male  26.000000      0      0            111369  30.0000  C148        C
890    male  32.000000      0      0            370376   7.7500   NaN        Q

[891 rows x 11 columns]
```

-3-Ticket- No la tendremos en cuenta por simplificar

[60]:
```
df.Ticket.value_counts()
```

[60]:
```
Ticket
347082             7
1601               7
CA. 2343           7
3101295            6
CA 2144            6
                  ..
PC 17590           1
17463              1
330877             1
373450             1
STON/O2. 3101282   1
Name: count, Length: 681, dtype: int64
```

-4-Cabin- No la tendremos en cuenta por falta de información

[61]:
```
df.Cabin.isnull().sum(), len(df)
```

[61]:
```
(np.int64(687), 891)
```

-5-Embarked

[62]:
```
df.Embarked.isnull().sum()
```

[62]:
```
np.int64(2)
```

[63]:
```
df.Embarked.value_counts()
```

[63]:
```
Embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```

[64]:
```
df["Embarked"] = df["Embarked"].fillna("S")
```

[65]:
```
df.Embarked.value_counts()
```

[65]:
```
Embarked
S    646
C    168
Q     77
Name: count, dtype: int64
```

[66]:
```
df.Embarked.isnull().sum()
```

[66]:
```
np.int64(0)
```

BORRAMOS del dataframe las columnas antes mencionadas

[67]:
```
df.head(2)
```

[67]:
```
   Survived  Pclass                                               Name  \
0         0       3                            Braund, Mr. Owen Harris
1         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th…

      Sex   Age  SibSp  Parch     Ticket     Fare Cabin Embarked
0    male  22.0      1      0  A/5 21171   7.2500   NaN        S
1  female  38.0      1      0   PC 17599  71.2833   C85        C
```

[68]:
```
df = df.drop(["Name", "Ticket", "Cabin"], axis=1)
df.head(2)
```

[68]:
```
   Survived  Pclass     Sex   Age  SibSp  Parch     Fare Embarked
0         0       3    male  22.0      1      0   7.2500        S
1         1       1  female  38.0      1      0  71.2833        C
```

Concepto de **datos categóricos**:

- columnas con strings hombre/mujer por ejemplo.
- columnas con strings con 3 opciones (“Embarked”)
- en el caso de Pclass 3 hace referencia a “tercera clase”
- y 3 no vale, más que 1, y más en este caso, cuya probabilidad de supervivencia es más baja.

[69]:
```
# pd.get_dummies()
```

[70]:
```
df_1 = pd.get_dummies(df, columns=["Sex", "Pclass", "Embarked"],␣
↪drop_first=True)
df_1.head()
```

[70]:

	Survived	Age	SibSp	Parch	Fare	Sex_male	Pclass_2	Pclass_3	Embarked_Q	Embarked_S
0	0	22.0	1	0	7.2500	True	False	True	False	True
1	1	38.0	1	0	71.2833	False	False	False	False	False
2	1	26.0	0	0	7.9250	False	False	True	False	True
3	1	35.0	1	0	53.1000	False	False	False	False	True
4	0	35.0	0	0	8.0500	True	False	True	False	True

[71]:
```
df = pd.get_dummies(df, columns=["Sex", "Pclass", "Embarked"], drop_first=True,␣
↪dtype=float)
df.head()
```

[71]:

	Survived	Age	SibSp	Parch	Fare	Sex_male	Pclass_2	Pclass_3	Embarked_Q	Embarked_S
0	0	22.0	1	0	7.2500	1.0	0.0	1.0	0.0	1.0
1	1	38.0	1	0	71.2833	0.0	0.0	0.0	0.0	0.0
2	1	26.0	0	0	7.9250	0.0	0.0	1.0	0.0	1.0
3	1	35.0	1	0	53.1000	0.0	0.0	0.0	0.0	1.0
4	0	35.0	0	0	8.0500	1.0	0.0	1.0	0.0	1.0

3.2 Escalado de los datos

Existen varias formas de hacer el escalado de datos. Normalmente no hay diferencias significativas, pero algunas veces sí.

Por abreviar, trataremos de mencionar 2 tipos (sklearn): * StandardScaler * MinMaxScaler

En nuestro caso, no daremos importancia a cuál es el mejor en este caso concreto. (Preporcesamiento)

[72]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.
 ↪StandardScaler.html
# https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.
 ↪MinMaxScaler.html
```

[73]:
```
# StandardScaler

# x-mean(x) / std(x)

df.Age = (df.Age - np.mean(df.Age, axis=0)) / (np.std(df.Age, axis=0))
df.Fare = (df.Fare - np.mean(df.Fare, axis=0)) / (np.std(df.Fare, axis=0))
df.head()
```

[73]:

	Survived	Age	SibSp	Parch	Fare	Sex_male	Pclass_2	Pclass_3	Embarked_Q	Embarked_S
0	0	-0.592481	1	0	-0.502445	1.0	0.0	1.0	0.0	1.0
1	1	0.638789	1	0	0.786845	0.0	0.0	0.0	0.0	0.0
2	1	-0.284663	0	0	-0.488854	0.0	0.0	1.0	0.0	1.0
3	1	0.407926	1	0	0.420730	0.0	0.0	0.0	0.0	1.0
4	0	0.407926	0	0	-0.486337	1.0	0.0	1.0	0.0	1.0

[74]:
```
df.describe()
```

[74]:

	Survived	Age	SibSp	Parch	Fare	Sex_male	Pclass_2	Pclass_3	Embarked_Q	Embarked_S
count	891.000000	8.910000e+02	891.000000	891.000000	8.910000e+02	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.232906e-16	0.523008	0.381594	3.987333e-18	0.647587	0.206510	0.551066	0.086420	0.725028
std	0.486592	1.000562e+00	1.102743	0.806057	1.000562e+00	0.477990	0.405028	0.497665	0.281141	0.446751
min	0.000000	-2.253155e+00	0.000000	0.000000	-6.484217e-01	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	-5.924806e-01	0.000000	0.000000	-4.891482e-01	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000e+00	0.000000	0.000000	-3.573909e-01	1.000000	0.000000	1.000000	0.000000	1.000000
75%	1.000000	4.079260e-01	1.000000	0.000000	-2.424635e-02	1.000000	0.000000	1.000000	0.000000	1.000000
max	1.000000	3.870872e+00	8.000000	6.000000	9.667167e+00					

```
max       1.000000     1.000000     1.000000     1.000000     1.000000
```

3.3 Obtención de X, y

[75]:
```
X = df.drop("Survived", axis=1)
X.head()
```

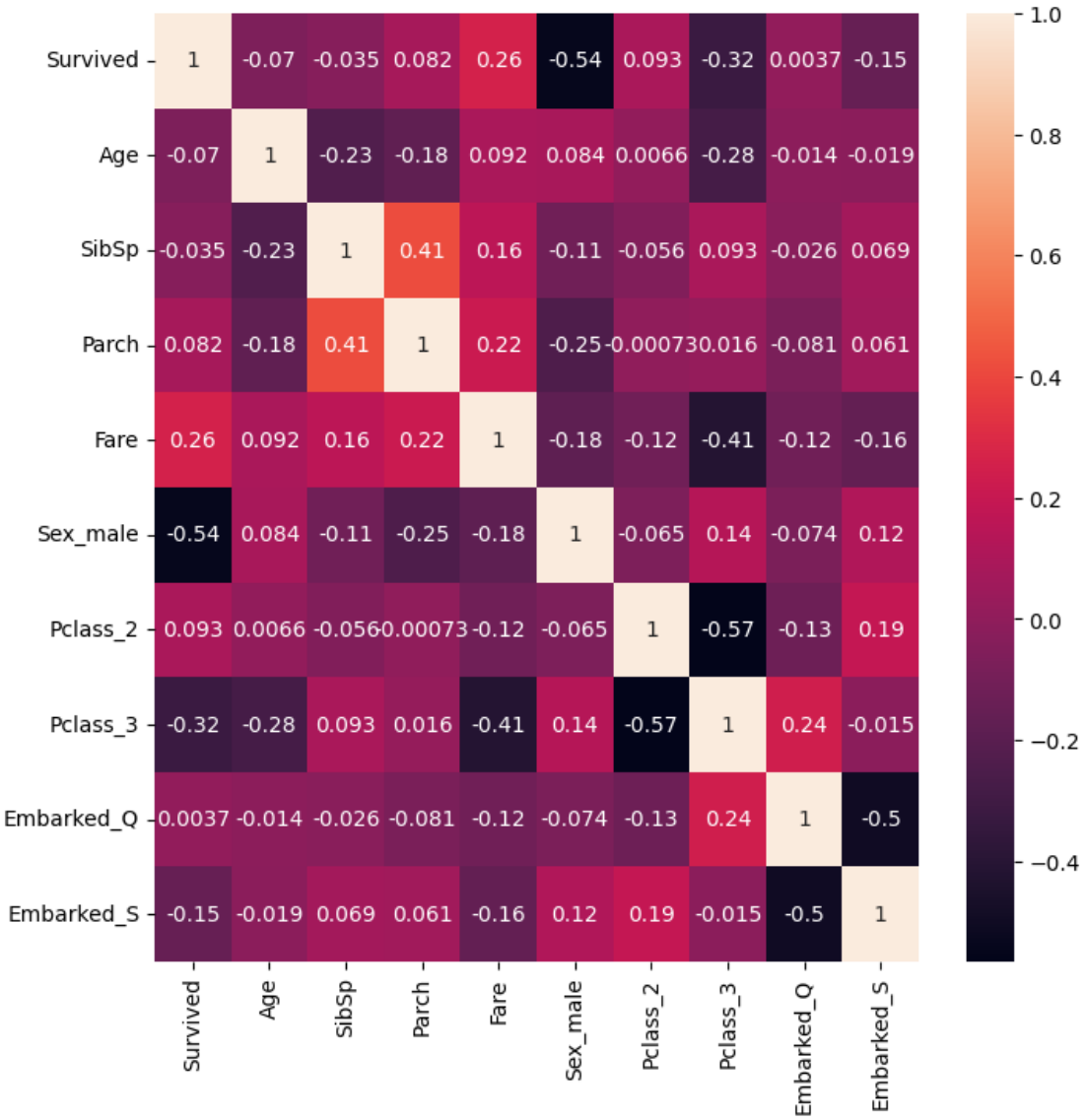
[75]:
```
        Age  SibSp  Parch      Fare  Sex_male  Pclass_2  Pclass_3  Embarked_Q  \
0 -0.592481      1      0 -0.502445       1.0       0.0       1.0         0.0
1  0.638789      1      0  0.786845       0.0       0.0       0.0         0.0
2 -0.284663      0      0 -0.488854       0.0       0.0       1.0         0.0
3  0.407926      1      0  0.420730       0.0       0.0       0.0         0.0
4  0.407926      0      0 -0.486337       1.0       0.0       1.0         0.0

   Embarked_S
0         1.0
1         0.0
2         1.0
3         1.0
4         1.0
```

[76]:
```
y = df["Survived"]
y.head()
```

[76]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

3.3.1 Heapmap

[77]:
```
plt.figure(figsize=(8,8))
sns.heatmap(df.corr(), annot=True)
plt.show()
```



Creado por:

Isabel Maniega