Universitat Oberta de Catalunya uoc.edu

Tipología y ciclo de vida de lo datos

Práctica 2: Limpieza y análisis de datos

Alumnas:

Erika Paola Martínez Soria

Eugenia Bezek

Índice

1. Descripción del dataset

El dataset que seleccionamos es el de Titanic. A partir de este conjunto de datos que representa distintas características de un grupo de 890 pasajeros del Titanic, se pretende por un lado determinar cuáles son las variables que más influyeron para que un pasajero sobreviva o no y también crear un modelo de regresión logística que dadas ciertas características de una persona nos devuelva la probabilidad de supervivencia. Tambíen podremos aplicar contrastes de hipótesis que nos ayudarán a resolver preguntas interesantes de la muestra para luego inferirlas a la población.

Es interesante resolver estas preguntas para entender qué se hizo mal en esa situación y modificar las medidas de rescate, prevención, salidas de emergencia, para que en caso de que otro barco de esas dimensiones se hunda no haya que lamentar víctimas fatales o sean las menos posibles.

El dataset y el código del mismo esta localizable en la siguiente dirección: https://github.com/eugeniabezek/Titanic_Data_Analysis

El conjunto de datos tiene 12 variables:

* **PassengerId**: ID to identity each passenger
* **Name**
* **Age**
* **Survived**: Survival: 0 = No, 1 = Yes
* **Pclass**: A proxy for socio-economic status (SES)
* **SibSp**: # of siblings / spouses aboard the Titanic
* **Parch**: # of parents / children aboard the Titanic:
* **Ticket**: Ticket number
* **Fare**: Passenger fare
* **Cabin**: Cabin number
* **Embarked**: Port of Embarkation.

2. Integración y selección de los datos de interés a analizar

Realizamos la lectura del fichero en formato CSV. El resultado devuelto por `read_csv()` es un `dataframe`.

```
df = pd.read_csv('titanic.csv', header = 0)
```

Obtenemos información sobre cada una de las 12 variables, cuantos valores no nulos existe y que tipo de variable es.

```
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
```

Obtenemos los valores estadísticos básicos de las variables numéricas del dataset.

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

De las variables del conjunto de datos, eliminamos las que no van a ser utilizadas porque no representan utilidad para las pruebas estadísticas.

```
df.drop(['PassengerId', 'Ticket', 'Cabin', 'Name'], axis=1,  inplace=True)
```

3. Limpieza de los datos

3.1. Tratamiento de valores nulos

Obervamos que la variable Age y la variable Embarked tienen valores nulos.

```
df.isnull().sum()
```

```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
Embarked      2
```

En este punto debemos decidir que hacer con los valores nulos. Podemos decidir eliminar los registros, pero eso significaría perder información. Así que decidimos, como mejor opción imputar los valores adecuados en los valores nulos de las dos variables incompletas.

En el caso de la variable Embarked, imputamos el valor de la moda en los dos valores nulos.

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

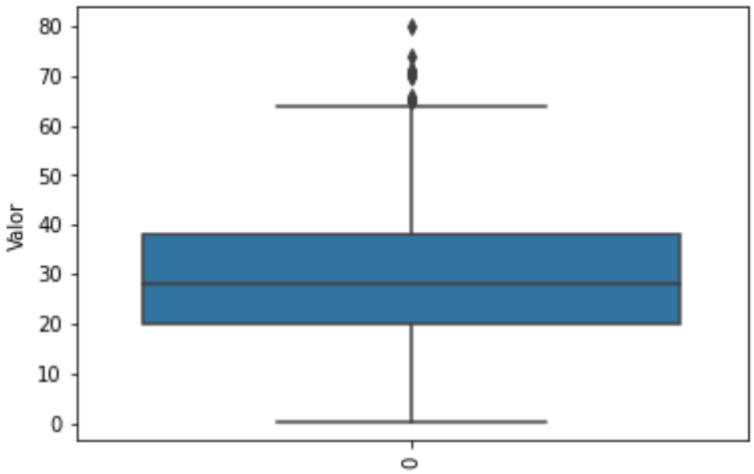
En el caso de la variable Age, debido a que su distribución es sesgada, utilizamos el valor de la mediana para imputar los 177 valores nulos.

```
df['Age'].fillna(df['Age'].median(), inplace=True)
```

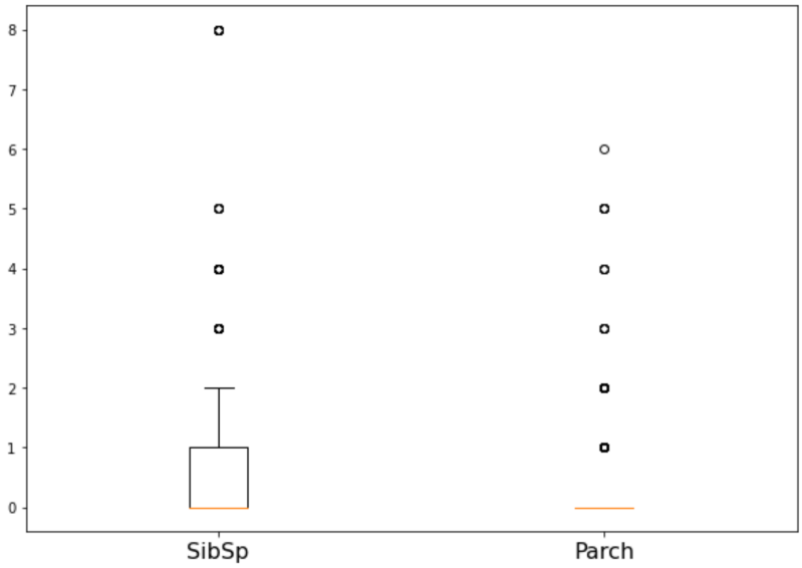
3.2. Tratamiento de valores extremos

Los valores extremos o *outliers* son aquellos que parecen no ser congruentes si los comparamos con el resto de los datos. Para visualizarlos decidimos utilizar diagramas de cajas o *boxplots*.

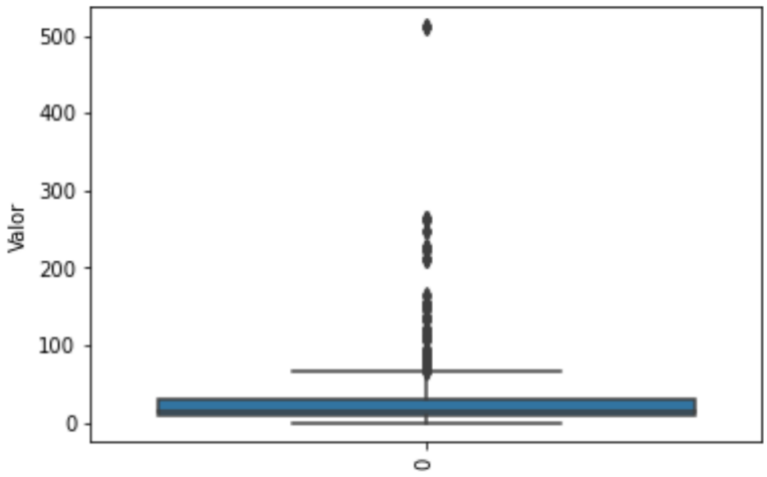
```
#Boxplot de la variable Age
age=df['Age']

sns.boxplot(data=age)
plt.xticks(rotation=90)
plt.ylabel('Valor')
plt.show()
```




```
#Boxplot de las variables SibSp y Parch
fig, ax = plt.subplots(figsize=(10,7))

ax.boxplot(df[["SibSp", "Parch"]])

labels = ["SibSp", "Parch"]
ax.set_xticklabels(labels, size=16)
```




```
#Boxplot de la variable Fare
fare=df['Fare']

sns.boxplot(data=fare)
plt.xticks(rotation=90)
plt.ylabel('Valor')
plt.show()
```




```
#Creamos una función que obtenga los outliers de la variables
def obtener_outlier_IQR(df):
    Q1=df.quantile(0.25)
    Q3=df.quantile(0.75)
    IQR=Q3-Q1
    df_final=df[((df<(Q1-1.5*IQR)) | (df>(Q3+1.5*IQR)))]
    return df_final
```

Analizamos en más detalle los valores extremos de cada variable.

```
for var in ["Age", "SibSp", "Parch", "Fare"]:
    variable=df[var]
    df_outliers = obtener_outlier_IQR(variable)

    print("El numero de outliers de la vairable {} es {}".format(var, len(df_outliers)))
    print("El valor minimo de entre los outliers es {} y el valor maximo es {}".format(df_outliers.min(), df_outliers.max()))
    print(" ")
```

```
El numero de outliers de la vairable Age es 11
El valor minimo de entre los outliers es 65.0 y el valor maximo es 80.0

El numero de outliers de la vairable SibSp es 46
El valor minimo de entre los outliers es 3 y el valor maximo es 8

El numero de outliers de la vairable Parch es 213
El valor minimo de entre los outliers es 1 y el valor maximo es 6

El numero de outliers de la vairable Fare es 116
El valor minimo de entre los outliers es 66.6 y el valor maximo es 512.3292
```

Los outliers de todas las variables son válidos, por lo que no tomamos ninguna acción con respecto a estos. En el caso de la variable Age una persona puede tener 80 años. En la variable SibSp, el número de hermanos o parejas de una persona puede ser 8. En el caso de Parch, el número de padres o hijos que un pasajero tiene si puede ser 6. En la variable Fare, el valor máximo outlier es 512, lo cualsi es posible ya que el ticket más costoso era £870 según https://www.cruisemummy.co.uk/titanic-ticket-prices/

4. Análisis de los datos.

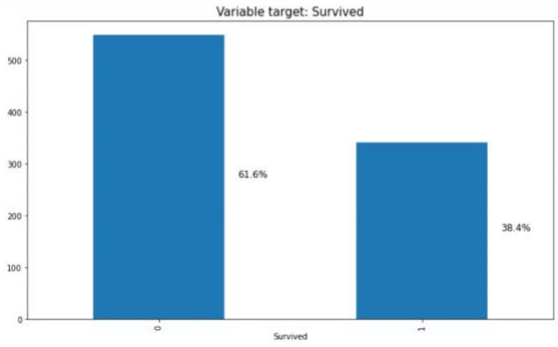
4.1. Selección de los grupos de datos que se quieren analizar/comparar (p. e., si se van a comparar grupos de datos, ¿cuáles son estos grupos y qué tipo de análisis se van a aplicar?)

Análisis de la variable **Survived**:

`Survived`

Indica si el pasajero sobrevivió (`1`) o no (`0`). Será nuestra variable a predecir. es decir la variable dependiente. Vemos que hay alrededor de un 38% de pasajeros que sobrevivieron.

```
ax = df.groupby(['Survived']).size().plot(kind = 'bar', stacked=True, figsize = (12,7))
total = len(df)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height()/total)
    x = p.get_x() + p.get_width() + 0.05
    y = p.get_y() + p.get_height()/2
    ax.annotate(percentage, (x, y), fontsize = 12)

plt.title('Variable target: Survived', fontsize = 15)
plt.show()
```

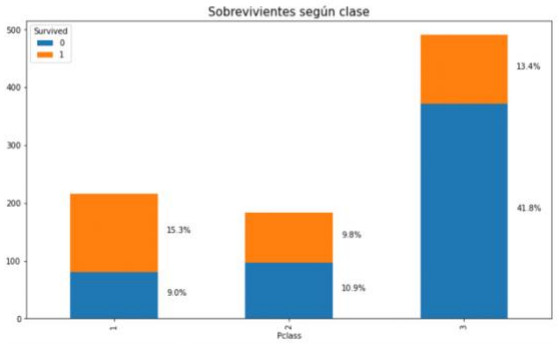


Análisis de la variable **Pclass**:

`Pclass`

Clase del tiquet. Es una variable categórica, con 3 valores diferentes correspondientes a la primera, la segunda y la tercera clase. Vemos que está correlacionada con la supervivencia: como más alta sea la clase, más supervivientes hay.

```
graficar_variable(df,df['Pclass'], df['Survived'])
plt.title('Sobrevivientes según clase', fontsize = 15)
plt.show()
```




```
crear_tabla(df,df['Pclass'])
```

Pclass	No_sobrevivio	Sobrevivio	Porcentaje_dataset
3	372	119	55.106622
1	80	136	24.242424
2	97	87	20.650954

```
df[['Pclass', 'Survived']].groupby(['Pclass'], as_index=False).mean()
```

	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.242363

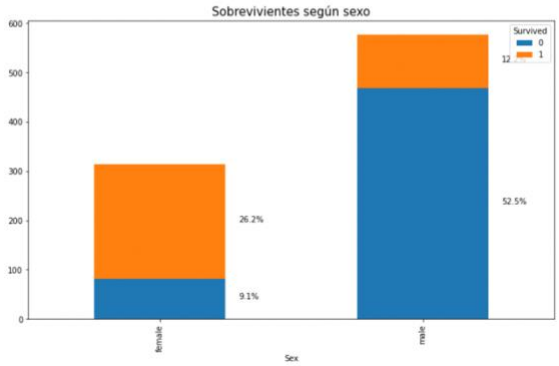
Teniendo en cuenta el gráfico y las tablas anteriores vemos que, los pasajeros de 3ra clase representan el 55% del dataset, seguido de los de primera clase que representan el 24,24% y por último la clase 2 que son el 20% restante. Dentro de cada grupo, vemos que el mayor porcentaje de sobrevivientes se da en el grupo de primera clase: un 63% de estos pasajeros sobrevivieron, mientras que sólo el 24% de los pasajeros de clase 3 lo hicieron.

Análisis de la variable **Sex**:

`Sex`

Sexo del pasajero. Es una variable categórica con dos valores, `male` y `female`. La convertiremos a binaria para nuestro análisis. Vemos que el 74% de mujeres sobrevivio, vs un 19% de hombres, por lo tanto, el sexo es una variable importante para determinar si un pasajero sobrevivio o no.

```
graficar_variable(df,df['Sex'], df['Survived'])
plt.title('Sobrevivientes según sexo', fontsize = 15)
plt.show()
```



```
crear_tabla(df,df['Sex'])
```

Sex	No_sobrevivio	Sobrevivio	Porcentaje_dataset
male	468	109	64.758698
female	81	233	35.241302

```
df[["Sex", "Survived"]].groupby(['Sex'], as_index=False).mean()
```

	Sex	Survived
0	female	0.742038
1	male	0.188908

Vemos que el 65% del dataset son paseros de sexo masculino, de los cuales sólo el 19% sobrevivió. El 35% restante son mujeres, y el porcentaje de sobrevivientes dentro de este grupo es del 74%.

Si tomamos toda la población, sobrevivieron un 26,2% de mujeres y un 12,2% de hombres.

Convertimos la variable en numérica:

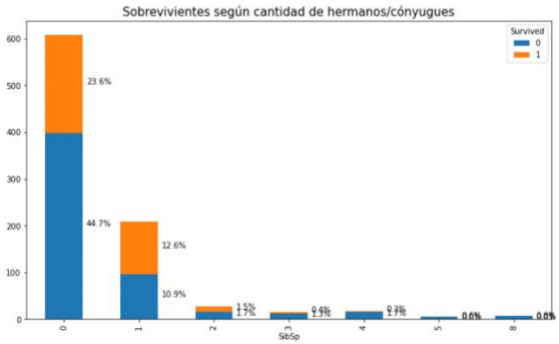
```
df['Sex'] = df['Sex'].replace(['male'],1)
df['Sex'] = df['Sex'].replace(['female'],0)
```

Análisis de la variable **SibSp**:

`SibSp`

Número de hermanos o cónyugues del pasajero dentro del Titanic. Es una variable numérica. Podemos ver que aquellos que viajaban con 1 hermano o cónyugue fueron el grupo que mas sobrevivientes tuvo.

```
graficar_variable(df,df['SibSp'], df['Survived'])
plt.title('Sobrevivientes según cantidad de hermanos/cónyugues', fontsize = 15)
plt.show()
```




```
crear_tabla(df,df['SibSp'])
```

SibSp	No_sobrevivio	Sobrevivio	Porcentaje_dataset
0	398.0	210.0	68.237935
1	97.0	112.0	23.456790
2	15.0	13.0	3.142536
4	15.0	3.0	2.020202
3	12.0	4.0	1.795735
5	5.0	NaN	NaN
8	7.0	NaN	NaN

```
df[["SibSp", "Survived"]].groupby(['SibSp'], as_index=False).mean()
```

	SibSp	Survived
0	0	0.345395
1	1	0.535885
2	2	0.464286
3	3	0.250000
4	4	0.166667
5	5	0.000000
6	8	0.000000

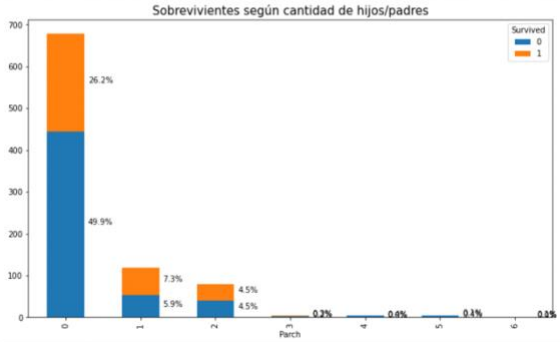
El 68% de los pasajeros no estaban acompañados ni por hermanos ni por sus cónyugues, luego hay un 23,4% que iban acompañados por unó solo de ellos, el resto de los grupos tienen porcentajes menor al 5% del total. Si analizamos los grupos por separado, de aquellos que iban con un hermano o cónyugue un 54% sobrevivió, siendo el grupo que le sigue en mayor porcentaje el que está formado por 2 hermanos/cónyugues, con un 46% de sobrevivientes. Si lo vemos a nivel general, tomando a todos los pasajeros, el 23,6% de los sobrevivientes fueron solos, y el 12.6% de sobrevivientes fue con un acompañante, los porcentajes de los siguientes grupos son menores al 2%.

Análisis de la variable **Parch**:

`Parch`

Número de padres e hijos del pasajero dentro del Titanic. Es una variable numérica. Aquellos que viajaban con tres padres o tres hijos, o una combinación de ellos que de tres, son los que más sobrevivieron.

```
graficar_variable(df,df['Parch'], df['Survived'])
plt.title('Sobrevivientes según cantidad de hijos/padres', fontsize = 15)
plt.show()
```




```
crear_tabla(df,df['Parch'])
```

Parch	No_sobrevivio	Sobrevivio	Porcentaje_dataset
0	445.0	233.0	76.094276
1	53.0	65.0	13.243547
2	40.0	40.0	8.978676
3	2.0	3.0	0.561167
5	4.0	1.0	0.561167
4	4.0	NaN	NaN
6	1.0	NaN	NaN

```
df[["Parch", "Survived"]].groupby(['Parch'], as_index=False).mean()
```

	Parch	Survived
0	0	0.343658
1	1	0.550847
2	2	0.500000
3	3	0.600000
4	4	0.000000
5	5	0.200000
6	6	0.000000

El 76% de los pasajeros del dataset fueron sin hijos y sin padres. Luego, el 13% fue acompañados por sólo un padre o un hijo, y hay un 8% que fueron con 2 de ellos. Luego, el resto de los grupos corresponden a menos del 1% de la población. Analizando cada grupo por separado, dentro del grupo mayoritario que fueron solos, el 34% sobrevivió. Dentro del grupo de 1 acompañante, el 55% sobrevivió y tomando el grupo de 2 acompañantes el 50% sobrevivió. Tomando los valores generales de toda la muestra, el mayor porcentaje de sobrevivientes es del grupo que fue sólo, con el 26,2% de sobrvivientes, seguido de aquellos que fueron con un padre o hijo, que corresponde al 7.3%.

Creación de la variable **FamilySize**:

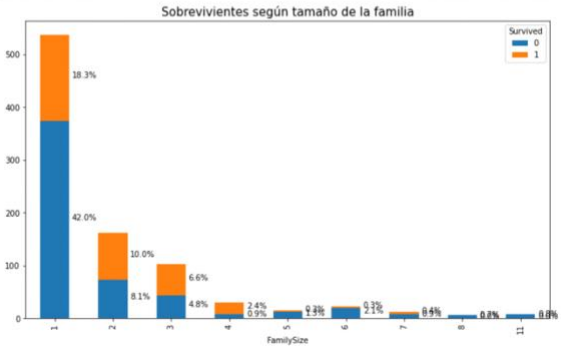
`FamilySize`

Con las variables `SibSp` y `Parch` podemos calcular el tamaño de la familia de un pasajero, que llamaremos `FamilySize`.

Las familias con 4 integrantes conforman el grupo que más sobrevivientes tuvo. Seguido por las de 3 y luego las de 2 integrantes.

```
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1
```

```
graficar_variable(df,df['FamilySize'], df['Survived'])
plt.title('Sobrevivientes según tamaño de la familia', fontsize = 15)
plt.show()
```



```
crear_tabla(df,df['FamilySize'])
```

FamilySize	No_sobrevivio	Sobrevivio	Porcentaje_dataset
1	374.0	163.0	60.269360
2	72.0	89.0	18.069585
3	43.0	59.0	11.447811
4	8.0	21.0	3.254770
6	19.0	3.0	2.469136
5	12.0	3.0	1.683502
7	8.0	4.0	1.346801
8	6.0	NaN	NaN
11	7.0	NaN	NaN

```
df[["FamilySize", "Survived"]].groupby(['FamilySize'], as_index=False).mean()
```

	FamilySize	Survived
0	1	0.303538
1	2	0.552795
2	3	0.578431
3	4	0.724138
4	5	0.200000
5	6	0.136364
6	7	0.333333
7	8	0.000000
8	11	0.000000

El 60% del dataset está formado por pasajeros que fueron solos, ya que el tamaño de su familia indica 1. El siguiente grupo en mayor porcentaje es el de familias formadas por dos integrantes, los cuales representan el 18% del dataset y luego los de 3 integrantes representan el 11,4%. El resto corresponden a grupo de menos del 5%. Si analizamos cada grupo por separado, del grupo de sólo 1 integrante, sobrevivió el 30%, pero a nivel general este grupo es el que más sobrevivientes tuvo, el 18,3%. Dentro del grupo de 2 integrantes, el 55% sobrevivió, pero a nivel general, esto corresponde al 10%.

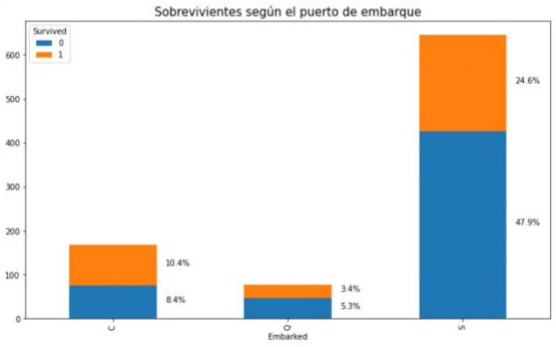
Como la variable FamilySize incluye las variables Parch y SibSp, eliminaremos estas últimas.

Análisis de la variable **Embarked**:

`Embarked`

Indica el puerto de embarque del pasajero. Es una variable categórica donde `C` indica que embarcó en Cherbourg, `Q` en Queenstown y `S` en Southampton.

```
graficar_variable(df,df['Embarked'], df['Survived'])
plt.title('Sobrevivientes según el puerto de embarque', fontsize = 15)
plt.show()
```




```
crear_tabla(df,df['Embarked'])
```

Embarked	No_sobrevivio	Sobrevivio	Porcentaje_dataset
S	427	219	72.502806
C	75	93	18.855219
Q	47	30	8.641975

```
df[['Embarked', 'Survived']].groupby(['Embarked'], as_index=False).mean()
```

	Embarked	Survived
0	C	0.553571
1	Q	0.389610
2	S	0.339009

El 72% de los pasajeros embarcaron en el puerto Southampton, el 19% en Cherbourg y el 9% restante en Queenstown. Analizando los grupos por separado, el 55% de los que embarcaron en Chercourg sobrevivieron, el 39% de los que embarcaron en Queenstown sobrevivieron y de los que lo hicieron en Southampton, sobrevivió el 34%.

La convertimos a variable numérica a través de variables dummy:

```
Embarked_dummy = pd.get_dummies(df['Embarked'])
df = pd.concat([df, Embarked_dummy], axis=1)
df.head()
```

	Survived	Pclass	Sex	Age	Fare	Embarked	FamilySize	C	Q	S
0	0	3	1	22.0	7.2500	S	2	0	0	1
1	1	1	0	38.0	71.2833	C	2	1	0	0
2	1	3	0	26.0	7.9250	S	1	0	0	1
3	1	1	0	35.0	53.1000	S	2	0	0	1
4	0	3	1	35.0	8.0500	S	1	0	0	1

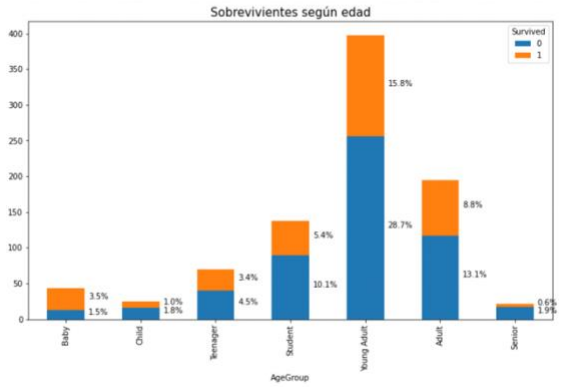
Análisis de la variable **Age**:

Age

Edad del pasajero. Categorizaremos la variable en 8 categorías según la edad que tenga. La categoría `Unknown` será la de aquellos pasajeros que no tengamos la edad.

```
df["Age"] = df["Age"].fillna(-0.5)
bins = [0, 5, 12, 18, 24, 35, 60, np.inf]
labels = ['Baby', 'Child', 'Teenager', 'Student', 'Young Adult', 'Adult', 'Senior']
df['AgeGroup'] = pd.cut(df["Age"], bins, labels = labels)
```

```
graficar_variable(df,df['AgeGroup'], df['Survived'])
plt.title('Sobrevivientes según edad', fontsize = 15)
plt.show()
```




```
crear_tabla(df,df['AgeGroup'])
```

AgeGroup	No_sobrevivio	Sobrevivio	Porcentaje_dataset
Young Adult	256	141	44.556678
Adult	117	78	21.885522
Student	90	48	15.488215
Teenager	40	30	7.856341
Baby	13	31	4.938272
Child	16	9	2.805836
Senior	17	5	2.469136

```
df[['AgeGroup', 'Survived']].groupby(['AgeGroup'], as_index=False).mean()
```

	AgeGroup	Survived
0	Baby	0.704545
1	Child	0.360000
2	Teenager	0.428571
3	Student	0.347826
4	Young Adult	0.355164
5	Adult	0.400000
6	Senior	0.227273

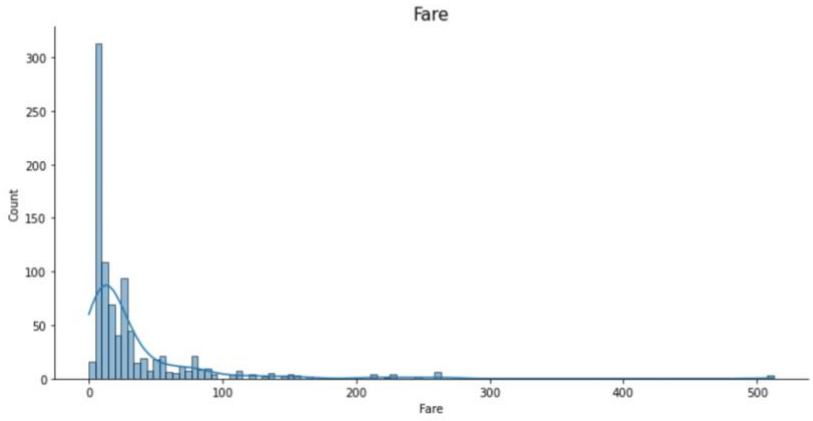
El 44,5% corresponde a la categoria "Young Adults" que son jóvenes de entre 24 y 35 años. El siguiente grupo mayoritario está formada por el grupo "Adult", que son los que tienen entre 35 y 60 años, y corresponden al 22% del dataset. Luego, el 15,5% corresponde a la categoría "Student" que son jóvenes de entre 18 y 24 años.
Tomando los grupos por separado, es de esperar que el mayor porcentaje de sobrevivientes lo tenga la categoría "Baby", seguido por los "Teenager".

Análisis de la variable **Fare**:

Fare

Indica el precio del pasaje. Es una variable numerica continua.

```
1 sns.displot(df['Fare'], kde=True, aspect = 2, height = 5)
2 plt.title('Fare', fontsize = 15)
3 plt.show()
```



Luego de realizar el análisis de todas las variables, que se detalla en el código adjunto, hemos decidido quedarnos con las siguientes variables: 'Pclass', 'Sex', 'Age', 'Fare', 'FamilySize', 'C', 'Q', 'S' y 'AgeGroup'.

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Con el objetivo de verificar la normalidad de los datos aplicaremos el test de Shapiro-Wilk.

Asumiendo como hipótesis nula que la población está distribuida normalmente, si el p-value es menor al nivel de significancia, que definiremos como alfa = 0.05, entonces la hipótesis nula es rechazada y se concluye que los datos no cuentan con una distribución normal. Si, por el contrario, el p-value es mayor a alfa entonces no se puede rechazar la H0 y se asume que los datos siguen una distribución normal.

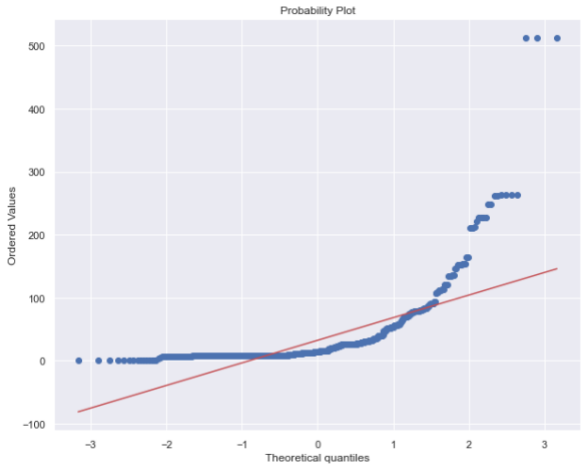
En nuestro caso validaremos la normalidad de la variable numérica Fare y Age.

Para el análisis de la variable “Fare” hemos obtenido un p-value menor que el valor de significancia alfa = 0.05, por lo tanto se rechaza la hipótesis nula y concluimos que la variable no tiene distribución normal.

```
from scipy.stats import shapiro

#Variable Fare
shapiro(df['Fare'])
```

```
ShapiroResult(statistic=0.5218914747238159, pvalue=1.0789998175301091e-43)
```

También lo evaluamos de manera gráfica con el gráfico cuantil-cuantil que compara los residuos del modelo con los valores de una variable que se distribuye normalmente:

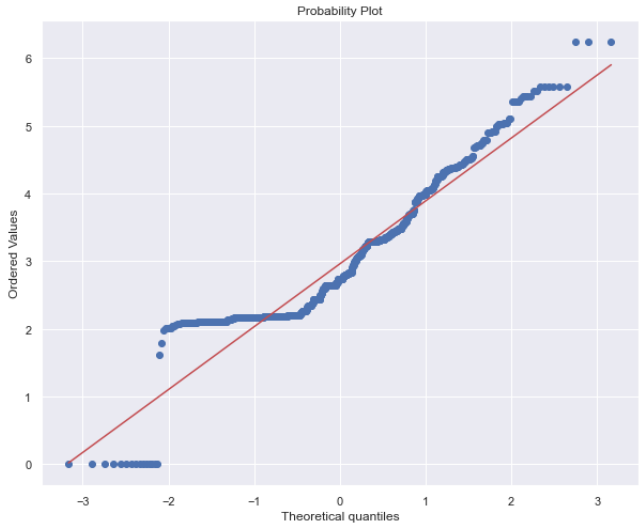


Luego aplicamos el logaritmo para ver si la distribución de la variable se acercaba más a una normal y el resultado fue nuevamente un p-value menor al valor de significancia pero gráficamente vemos que mejoró:

```
df['Fare'] = np.log1p(df['Fare'])
#Variable Fare
shapiro(df['Fare'])
```

```
ShapiroResult(statistic=0.9152467846870422, pvalue=6.642045243763613e-22)
```

```
stats.probplot(df['Fare'], dist="norm", plot=pylab)
plt.show()
```

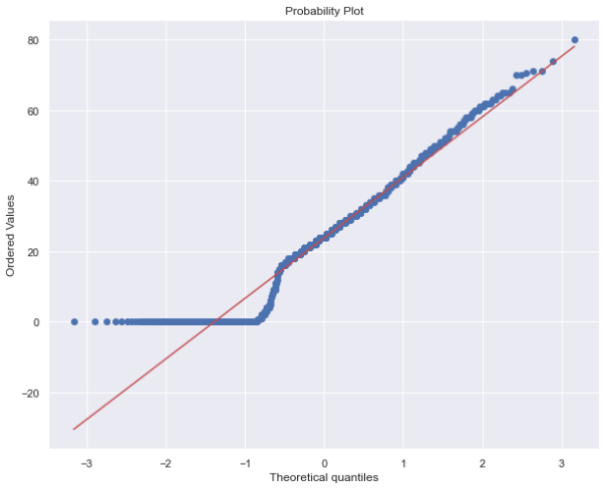


Pero, teniendo en cuenta el Teorema del Límite Central, que indica que la media de una muestra es cada vez más normal a medida que aumenta la cantidad de observaciones.
Se considera suficientemente grande a más de 30 observaciones, en este caso contamos con 890 valores, por lo que se podría considerar que la variable sigue una distribución normal.

A continuación hicimos el mismo análisis para la variable “Age”, donde también obtuvimos un valor de p menor a alfa, dando como conclusión que no sigue una distribución normal, pero se podría aplicar el mismo teorema del límite central para considerarla normal.

```
#Variable Age
shapiro(df['Age'])
```

```
ShapiroResult(statistic=0.9541045427322388, pvalue=4.650938269969563e-16)
```



Para el análisis de la homogeneidad de varianzas utilizaremos el test de Fligner-Killeen, ya que es un test no paramétrico, que se utiliza cuando las variables no siguen una distribución normal, dado el resultado obtenido anteriormente.
La hipótesis nula H0 asume igualdad de varianzas en los diferentes grupos de datos, por lo que p-values menores al nivel de significancia indicarán heterocedasticidad.

En este caso, estudiaremos esta homogeneidad entre los grupos conformados por mujeres y por hombres. Obtuvimos un p-value menor al valor de significancia, por lo tanto se rechaza la hipótesis nula y concluimos con una confianza del 95% que las varianzas de las muestras son distintas, es decir no hay la misma varianza en la muestra de hombres que de mujeres:

```
stats.fligner(df['Survived'], df['Sex'], center='mean')
```

```
FlignerResult(statistic=10.746465944041198, pvalue=0.0010447864597771198)
```

Luego, aplicamos el mismo método para evlauar la homogeneidad de varianzas entre las muestras conformados por las distintas clases, en este caso también el p-value fue menor al nivel de significancia por lo que las muestras tienen varianzas distintas:

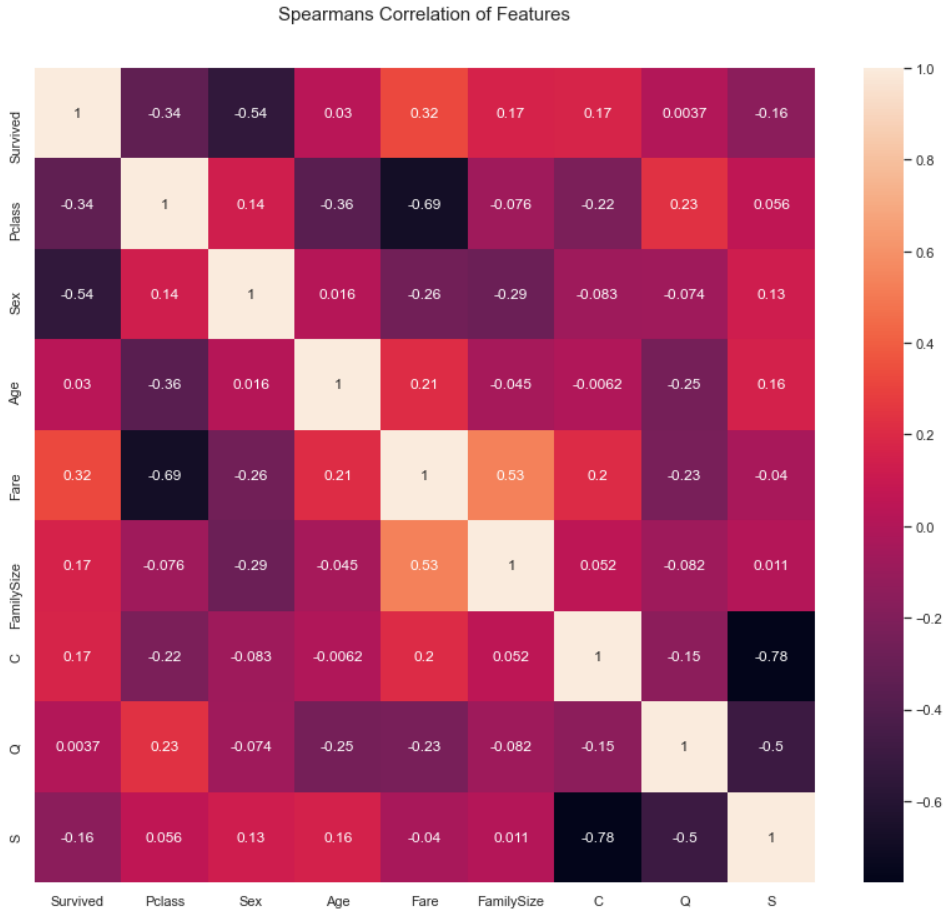
```
stats.fligner(df['Survived'], df['Pclass'], center='mean')
```

```
FlignerResult(statistic=585.8745259607911, pvalue=1.9781661175685402e-129)
```

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Primera pregunta a resolver: *¿Qué variables cuantitativas influyen más en la supervivencia de un pasajero?*

Para entender qué variable es más influyente calcularemos el coeficiente de correlación de Spearman ya que necesitamos un método no paramétrico por la falta de normalidad de las variables.



Analizando la matriz de correlación, vemos que la variable con mayor correlación con 'Survived' es Sex, con un coeficiente de -0.54, seguido de Pclass con un coeficiente de -0.34 y Fare con un coeficiente de 0.32:

```
coef, p = spearmanr(df['Survived'], df['Sex'])
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('La variable Survived y Sex no están correlacionadas (se acepta H0) p=%.3f' % p)
else:
        print('La variable Survived y Sex están correlacionadas (se rechaza H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.543
La variable Survived y Sex están correlacionadas (se rechaza H0) p=0.000
```

```
coef, p = spearmanr(df['Survived'], df['Pclass'])
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('La variable Survived y Pclass no están correlacionadas (se acepta H0) p=%.3f' % p)
else:
        print('La variable Survived y Pclass están correlacionadas (se rechaza H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.340
La variable Survived y Pclass están correlacionadas (se rechaza H0) p=0.000
```

```
coef, p = spearmanr(df['Survived'], df['Fare'])
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('La variable Survived y Fare no están correlacionadas (se acepta H0) p=%.3f' % p)
else:
        print('La variable Survived y Fare están correlacionadas (se rechaza H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: 0.324
La variable Survived y Fare están correlacionadas (se rechaza H0) p=0.000
```

Segunda pregunta a resolver: *¿La proporción de sobrevivientes que embarcaron en el puerto Cherbourg es mayor que la cantidad de sobrevivientes que embarcaron en el puerto Queenstown?*

La prueba estadística que se aplicará para resolver esta pregunta consistirá en un contraste de hipótesis sobre la proporción de dos muestras para determinar si la supervivencia es mayor si el embarque fue en Cherbourg comparado con Queenstown.

Una de las muestras será la cantidad de sobrevivientes que embarcaron en el puerto Cherbourg, y la otra muestra será la cantidad de sobrevivientes que embarcaron en Queenstown.

La hipótesis nula es que las proporciones de las muestras son iguales, la hipótesis alternativa es la proporción de sobrevivientes que embarcaron en Cherbourg es mayor que los que embarcaron en Queenstown.
Para evaluar esto aplicaremos un test de proporción z de dos muestras.
Dado que obtuvimos un p-value menor al valor de significancia, podemos concluir que, con un nivel de confianza del 95%, la proporción de todos los sobrevivientes del Titanic que embarcaron en el puerto Cherbourg es mayor la proporción de sobrevivientes que embarcaron en el puerto Queenstown.

```
from statsmodels.stats.proportion import proportions_ztest
alfa = 0.05
sample_C, sample_size_C = (df[(df['Embarked'] == 'C') & (df['Survived'] == 1)]['Embarked'].size, df[(df['Embarked'] == 'C') | (df['Embarked'] == 'Q') & (df['Survived'] == 1)]['Embarke
sample_Q, sample_size_Q = (df[(df['Embarked'] == 'Q') & (df['Survived'] == 1)]['Embarked'].size, df[(df['Embarked'] == 'C') | (df['Embarked'] == 'Q') & (df['Survived'] == 1)]['Embarke

samples = np.array([sample_C, sample_Q], dtype = object)
samples_size = np.array([sample_size_C, sample_size_Q], dtype = object)

stat, p_value = proportions_ztest(count=samples, nobs=samples_size,  alternative='larger')

print('z_stat: %0.3f, p_value: %0.3f' % (stat, p_value))

if p_value > alfa:
    print ("El p-value es mayor al valor de significancia de 0.05, por lo que no se puede rechazar la hipótesis nula y se concluye que las proporciones de las muestras son iguales")
else:
    print ("El p-value es menor al valor de significancia  de 0.05, por lo que se rechaza la hipótesis nula y se concluye que la muestra 1 es mayor que la muestra 2")
```

```
z_stat: 6.842, p_value: 0.000
El p-value es menor al valor de significancia  de 0.05, por lo que se rechaza la hipótesis nula y se concluye que la muestra 1 es mayor que la muestra 2
```

Tercera pregunta a resolver: *¿El valor promedio del ticket de los pasajeros que sobrevivieron es mayor que el valor promedio del ticket de los pasajeros que no sobrevivieron?*

La prueba estadística que se aplicará para resolver esta pregunta consistirá en un contraste de hipótesis sobre la media de dos muestras para determinar si el precio promedio del ticket de los pasajeros que sobrevivieron es mayor al precio promedio del ticket de aquellos que no lo hicieron.

La hipótesis nula es que las medias de las muestras son iguales, la hipótesis alternativa es que el precio promedio del ticket de los sobrevivientes es mayor que el de los no sobrevivientes.

La muestra 1 está formada por el valor de los tickets de aquellos pasajeros que sobrevivieron, y la muestra 2 está formada por el valor de los tickets de los pasajeros que no sobrevivieron. Como vimos anteriormente, la variable 'Fare' al tener más de 30 observaciones podemos considerarla normal basandonos en el Teorema del Limite Central, por lo que usaremos el t-test para realizar el análisis.
Teniendo en cuenta que obtuvimos un p-value menor al valor alfa, podemos concluir con un 95% de confianza que el valor del ticket promedio que pagaron los pasajeros que sobrevivieron es mayor al valor del ticket promedio que pagaron aquellos que no sobrevivieron, esto puede tener que ver con las medidas de seguridad diferenciales para las distintas clases y la prioridad que se les dio a unos frente a otros:

```
sample_1 = df[(df['Survived'] == 1)]['Fare']
sample_2 = df[(df['Survived'] == 0)]['Fare']

stat, p_value = stats.ttest_ind(sample_1, sample_2, equal_var=False, alternative='greater')

print('z_stat: %0.3f, p_value: %0.3f' % (stat, p_value))

if p_value > alfa:
    print ("El p-value es mayor al valor de significancia de 0.05, por lo que no se puede rechazar la hipótesis nula y se concluye que las medias de las muestras son iguales")
else:
    print ("El p-value es menor al valor de significancia  de 0.05, por lo que se rechaza la hipótesis nula y se concluye que la media de la muestra 1 es mayor que la media de muestra
```

```
z_stat: 10.121, p_value: 0.000
El p-value es menor al valor de significancia  de 0.05, por lo que se rechaza la hipótesis nula y se concluye que la media de la muestra 1 es mayor que la media de muestra 2
```

Cuarto análisis estadístico: Modelo de regresión logística

Se utiliza el criterio de información de Akaike para determinar qué variables deben introducirse al modelo. Se considera la variable más importante aquella que da lugar al cambio más grande en el índice AIC, el cual está basado en el deviance y el número de parámetros del modelo. Primero se ajusta el modelo con solo la constante y se compara el modelo que resulta de introducir cada una de las variables. La que de un valor de AIC más pequeño es considerada la más importante y se introduce en el modelo.
Teniendo en cuenta este criterio, el mejor modelo será aquel que explique la mayor parte de la varianza utilizando la menor cantidad de variables independientes posible. Luego de probar distintas combinaciones, hemos seleccionado el modelo formado por las siguientes variables independientes: “Pclass”, “Sex”, “Age”, “Fare”, “Embarked dummy” y “Family Size”, el valor de AIC que obtuvimos es de 566:

```
#Separamos la variable objetivo de las demas:
y = df['Survived']
x = df[['Pclass', 'Sex', 'Age','Fare', 'C','Q','S','FamilySize']]
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=0)
```

```
import statsmodels.api as sm
model = sm.Logit(y_train, X_train).fit()
print("El AIC del modelo es: ",model.aic)
```

```
Optimization terminated successfully.
         Current function value: 0.442070
         Iterations 6
El AIC del modelo es:  566.8197667783448
```

Además para entender qué tan bien clasificaba el modelo seleccionado, realizamos la matriz de confusión utilizando los datos que habíamos guardado para el “test” , que

corresponden al 30% de los datos originales, y obtuvimos un valor de accuracy de 0.8%:

```
from sklearn.metrics import confusion_matrix, accuracy_score

x = df[['Pclass', 'Sex', 'Age','Fare', 'C','Q','S','FamilySize']]
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=0)
model = sm.Logit(y_train, X_train).fit()
y_pred_proba = model.predict(X_test)
y_pred = list(map(round, y_pred_proba))

matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(matrix, annot=True, fmt="d", cmap='Blues', square=True)
plt.title('Matriz de confusión del set de prueba', fontsize = 15)
plt.xlabel("predicción")
plt.ylabel("real")
plt.show()

# accuracy score of the model
print('Test accuracy = ', accuracy_score(y_test, y_pred))
```

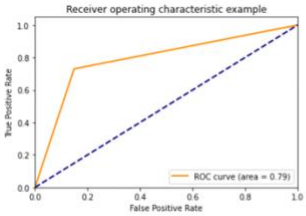
```
Optimization terminated successfully.
         Current function value: 0.442070
         Iterations 6
```


```
Test accuracy =  0.8059701492537313
```

Hemos obtenido un valor de accuracy de 0.8%, este es un indicadot de qué tan bien clasifica el modelo, está dado por: TN+TP/Total.

A continuación se muestra la curva ROC

```
from sklearn.metrics import roc_curve, auc

fpr, tpr, thresholds = roc_curve(y_test, y_pred)
roc_auc = auc(fpr, tpr)
plt.figure()
lw = 2
plt.plot(fpr, tpr, color='darkorange', lw=lw, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
```



El área debajo de la curva representa la habilidad del modelo para discriminar entre la clase negativa y la positiva. Un área de 1 representa un modelo que ha hecho todas las predicciones perfectas, mientras que un área de 0.5 representa el rendimiento de un modelo aleatorio. En este caso, el area es 0,79, es decir que el modelo clasifica bien casi el 80% de los casos.

Una vez que seleccionamos el modelo, pasamos a predecir la probabilidad de sobrevivir o no de pasajeros con distintas características:

```
Xtest = [3, 0, 25, 5.5, 0, 0, 1, 1]
ypred = model.predict(Xtest)
print("La probabilidad de que una mujer de 25 años, que viajó en tercera clase con su marido, se embarcó en el puerto S y pagó un ticket de 5.5 USD, sobreviva es: ", ypred*100)
```

```
La probabilidad de que una mujer de 25 años, que viajó en tercera clase con su marido, se embarcó en el puerto S y pagó un ticket de 5.5 USD, sobreviva es:  [83.72388289]
```

```
Xtest = [3, 1, 25, 5.5, 0, 0, 1, 1]
ypred = model.predict(Xtest)
print("La probabilidad de que un hombre de 25 años, que viajó en tercera clase con su mujer, se embarcó en el puerto S y pagó un ticket de 5.5 USD, sobreviva es: ", ypred*100)
```

```
La probabilidad de que un hombre de 25 años, que viajó en tercera clase con su mujer, se embarcó en el puerto S y pagó un ticket de 5.5 USD, sobreviva es:  [24.7365451]
```

```
Xtest = [1, 0, 25, 10.5, 0, 0, 1, 1]
ypred = model.predict(Xtest)
print("La probabilidad de que una mujer de 25 años, que viajó en primera clase con su marido, se embarcó en el puerto S y pagó un ticket de 10.5 USD, sobreviva es: ", ypred*100)
```

```
La probabilidad de que una mujer de 25 años, que viajó en primera clase con su marido, se embarcó en el puerto S y pagó un ticket de 10.5 USD, sobreviva es:  [99.24384766]
```

```
Xtest = [1, 1, 25, 10.5, 0, 0, 1, 1]
ypred = model.predict(Xtest)
print("La probabilidad de que un hombre de 25 años, que viajó en primera clase con su mujer, se embarcó en el puerto S y pagó un ticket de 10.5 USD, sobreviva es: ", ypred*100)
```

```
La probabilidad de que un hombre de 25 años, que viajó en primera clase con su mujer, se embarcó en el puerto S y pagó un ticket de 10.5 USD, sobreviva es:  [89.34574155]
```

```
Xtest = [1, 1, 55, 15, 0, 0, 1, 1]
ypred = model.predict(Xtest)
print("La probabilidad de que un hombre de 55 años, que viajó en primera clase con su mujer, se embarcó en el puerto S y pagó un ticket de 15 USD, sobreviva es: ", ypred*100)
```

```
La probabilidad de que un hombre de 55 años, que viajó en primera clase con su mujer, se embarcó en el puerto S y pagó un ticket de 15 USD, sobreviva es:  [90.09409919]
```

```
Xtest = [0, 1, 55, 15, 0, 0, 1, 1]
ypred = model.predict(Xtest)
print("La probabilidad de que una mujer de 55 años, que viajó en primera clase con su marido, se embarcó en el puerto S y pagó un ticket de 15 USD, sobreviva es: ", ypred*100)
```

```
La probabilidad de que una mujer de 55 años, que viajó en primera clase con su marido, se embarcó en el puerto S y pagó un ticket de 15 USD, sobreviva es:  [95.72692961]
```

```
Xtest = [1, 1, 1, 10, 1, 0, 0, 2]
ypred = model.predict(Xtest)
print("La probabilidad de que un niño de 1 año, que viajó en primera clase con sus padres, se embarcó en el puerto C y pagó un ticket de 10 USD, sobreviva es: ", ypred*100)
```

```
La probabilidad de que un niño de 1 año, que viajó en primera clase con sus padres, se embarcó en el puerto C y pagó un ticket de 10 USD, sobreviva es:  [95.42725914]
```

```
Xtest = [3, 1, 1, 7, 1, 0, 0, 2]
ypred = model.predict(Xtest)
print("La probabilidad de que un niño de 1 año, que viajó en tercera clase con sus padres, se embarcó en el puerto C y pagó un ticket de 7 USD, sobreviva es: ", ypred*100)
```

```
La probabilidad de que un niño de 1 año, que viajó en tercera clase con sus padres, se embarcó en el puerto C y pagó un ticket de 7 USD, sobreviva es:  [59.23133604]
```

Según los resultados que vimos, podemos sacar algunas conclusiones:

- Si el pasajero viajó en tercera clase, la probabilidad de sobrevivir es menor que si viajo en primera
- Si el pasajero era hombre, la probabilidad de sobrevivir es menor que si era mujer.
- Si el pasajero era hombre de tercera clase la probabilidad de sobrevivir es muy baja.
- Si el pasajero era un niño o bebé las probabilidades de sobrevivir son más altas.

5. Representación de los resultados a partir de tablas y gráficas

Este apartado se puede responder a lo largo de la práctica, sin necesidad de concentrar todas las representaciones en este punto de la práctica.

6. Resolución del problema

A partir de los resultados obtenidos, ¿cuáles son las conclusiones?. ¿Los resultados permiten responder al problema?

Para resolver la pregunta de cuál es la variable que más influye al momento de determinar si un pasajero sobrevivió o no al hundimiento del Titanic aplicamos el test de Spearman, que es una alternativa no paramétrica a la correlación de Pearson para medir el grado de dependencia entre variables.
Pudimos observar que la principal dependencia aparecía entre la variable Survived y Sex, con un coeficiente de -0,54, y luego seguía la dependencia moderada/débil entre Survived y Pclass (-0.34) y entre Survived y Fare (0.32).

La segunda pregunta que queríamos resolver era si podíamos inferir a partir de esta muestra que la proporción de pasajeros sobrevivientes que embarcaron en el puerto Cherbourg fue mayor a la proporción de sobrevivientes que embarcaron en el puerto Queenstown.

Estamos frente a un test de proporción de dos muestras bilateral por la derecha.
Para resolverla utilizamos el test Z de proporción de dos muestras independientes, seteando la alternativa "greater" ya que nuestra hipótesis alternativa es que la proporción de la primera muestra (los sobrevivientes que embarcaron en C) era mayor que la segunda (los sobrevivientes que embarcaron en Q). Pudimos concluir con un nivel de confianza del 95%, que la hipótesis alternativa era cierta.

Luego nos planteamos la pregunta de si el valor promedio del ticket que pagaron aquellos pasajeros que sobrevivieron era mayor al que pagaron aquellos que no sobrevivieron. En este caso estamos ante un test de contraste de hipótesis de dos muestras independientes sobre la media, y particularmente este caso se trata de un contraste biilateral por la derecha (ya que la hipótesis alternativa indica que la media es mayor).
Para resolverlo utilizamos el T-test para las medias de dos muestras independientes, obtuvimos como resultado que con un 95% de confianza, el valor del ticket promedio que pagaron los pasajeros que sobrevivieron es mayor al valor del ticket promedio que pagaron aquellos que no sobrevivieron.

Por último, creamos un modelo de regresión logística para poder predecir si un pasajero sobrevivirá o no teniendo en cuenta algunas variables independientes. Para poder determinar qué variables utilizar probamos distintas combinaciones y nos quedamos con el modelo que devolvió el menor AIC. Luego, pudimos utilizar este modelo para predecir la probabilidad de que un pasajero sobreviva dadas ciertos valores de: Sexo, Clase, Valor del ticket, Tamaño de la familia y el puerto de embarque.

7. Código

Adjuntamos código en Python en donde se ha realizado la limpieza, análisis y representación de los datos.

El dataset y el código del mismo esta localizable en la siguiente dirección:
https://github.com/eugeniabezek/Titanic_Data_Analysis

Contribuciones	Firma
Investigación previa	Erika Paola Martínez Soria Eugenia Bezek
Redacción de las respuestas	Erika Paola Martínez Soria Eugenia Bezek
Desarrollo código	Erika Paola Martínez Soria Eugenia Bezek