Trabajo práctico N°1-Industrias X.0 (92.26)

Integrantes:

Nombre	Padrón
Erman Octavio	101328
Ferrer Vieyra, Enrique	95992
Casado de Achaval, Blas	101082
Fraga Diaz de Vivar, Joaquín	99171
Delaney, Federico Luis	99550

Corrector: Marco Luquer

El objetivo del presente trabajo es a través del lenguaje Python, importar un dataset, realizarle un análisis exploratorio, luego una limpieza y un análisis de los datos resultantes para luego tratar de armar un modelo tal que logre predecir según los datos dados, si el tripulante sobrevivió o no.

En cuanto al primer punto de la exploración, mediante un pd.read.csv se lee el set de datos y se lo importa a la consola de Python. Aplicamos las funciones describe e info para estar al tanto de que variables manejamos, cuál es su tipo y qué características tiene.

```
[891 rows x 12 columns]
<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
```

1) Como se puede ver, el data set contiene 12 columnas con hasta 889 valores en algunos casos, y hay que tener en cuenta que en algunos casos para ciertas variables hay datos faltantes que deben ser o completados o eliminados del dataset. En este caso, se observa que 4 variables son descriptivas y 8 son variables numéricas, además en algunas columnas aparecen valores nulos, y en particular en el caso de la columna de Cabin, aproximadamente el 78% de los valores son nulos.

Aplicando el siguiente código se obtienen los siguientes resultados, teniendo en cuenta que los 1 corresponden a las personas que efectivamente sobrevivieron:

```
dataframe = pd.read_csv("tp.csv")
print(dataframe)
dataframe.describe()
dataframe.info()
Sobrevivientes= dataframe["Survived"].value_counts()
print("Personas que sobreviven y personas que no")
print(Sobrevivientes)
Sexo= dataframe["Sex"].value_counts()
print(Sexo)
```

```
Personas que sobreviven y personas que no
0    549
1    342
```

```
Cantidad de mujeres y hombres a bordo
male      577
female    314
```

2) Para la segunda parte de limpieza, en este caso se decidió hacerlo de dos maneras distintas, una consta de tratar de reemplazar los valores nulos según ciertos criterios específicos. Por ejemplo, para el caso de las edades se ha calculado el valor medio y desvío estándar para completar con números aleatorios que sigan una distribución normal.

```
Agemean = dataframe["Age"].mean()
ages = dataframe["Age"].dropna()
Agestdev = statistics.stdev(ages)
print(Agemean)
print(Agestdev)
```

```
dataframe["Age"].replace(np.nan,np.random.normal(loc=Agemean,scale=Agestdev), inplace=True)
print(dataframe)
dataframe.info()
```

Luego, para el caso de los embarcados, solamente faltan dos datos, y por lo tanto, se decidió buscar al momento de la salida del Titanic las poblaciones de las 3 distintas ciudades donde recogió pasajeros, y completar esos dos datos con aquella ciudad que mayor población tenia, en este caso fue la ciudad de Southampton.

```
dataframe["Embarked"].fillna("S", inplace=True)
print(dataframe)
```

Por último, para el caso de las cabinas, al ser un dato muy específico y no disponer de un manifiesto para tener la información sobre que cabinas estaban disponibles, se decidió reemplazar los valores nulos por “No Cabin”.

```
dataframe["Cabin"].fillna("No Cabin", inplace=True)
print(dataframe)
```

Por otra parte, la otra alternativa es la de eliminar aquellos valores nulos del dataset, pero previo a eso, como casi todos los valores nulos corresponden a la columna de Cabin, que a criterio del grupo no es imprescindible para la predicción que se debe hacer.

```
df = pd.read_csv("tp.csv")
del df["Cabin"]
df = df.dropna()
print("informacion de DF")
df.info()
```

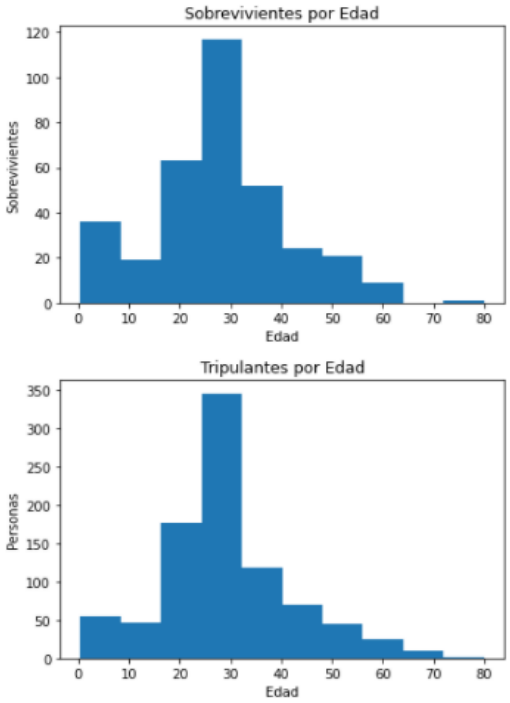
Al utilizar la función info queda el siguiente dataset:

```
 #   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          891 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        891 non-null    object
 11  Embarked     891 non-null    object
```

3) Para la visualización de datos, se analizan ciertas variables que puedan ser influyentes en si los pasajeros sobrevivieron o no, para esto se realizan filtraciones del dataset y se las compara una a una con la columna de “Survived” y luego se lo compara a esos datos realizándole un filtro con solo las personas que realmente sobrevivieron, de esta forma al visualizar los gráficos se puede obtener si hay o no relación entre las variables.

En primer lugar, se realizan las comparaciones habiendo completado los espacios nulos con los criterios previamente explicados

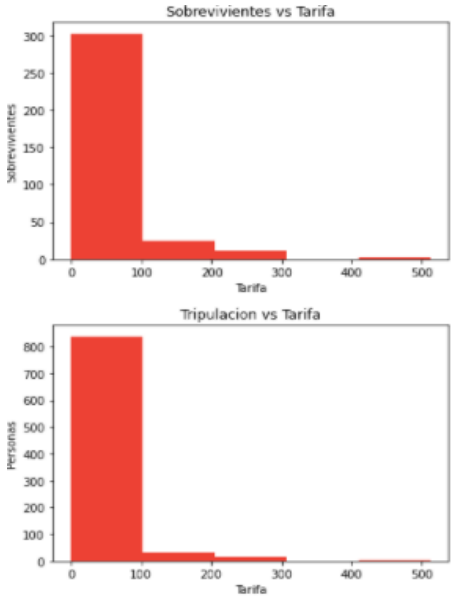
Para el caso de la edad relacionada con la tasa de supervivencia:



En este caso, se puede visualizar claramente que aquellas personas que se encuentran entre los 0 y 10 años tienen una altísima tasa de supervivencia, mientras que menos del 50% de las

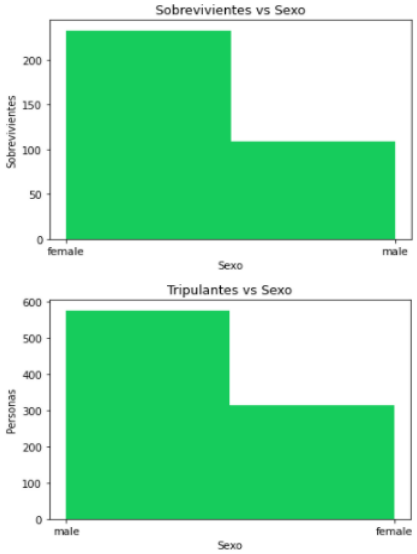
personas de entre 20 y 40 años sobreviven. Además, sorpresivamente la vasta mayoría de los tripulantes de entre 70 y 80 años también sobreviven. De esta manera, mediante este análisis se establece una fuerte relación entre la edad y la posibilidad de supervivencia.

En cuanto a la tarifa por ejemplo, se obtuvieron los siguientes resultados:



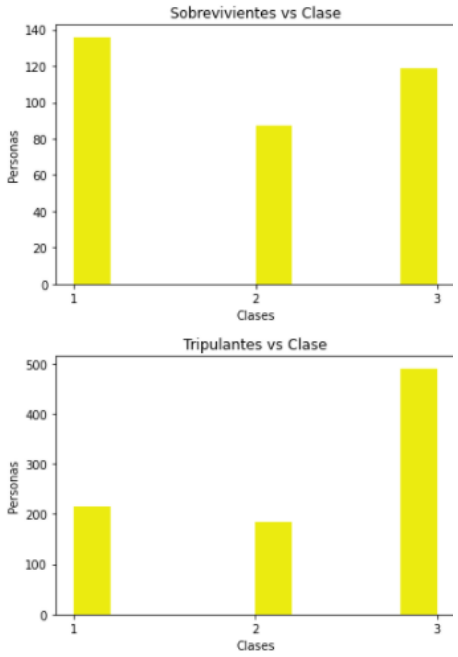
En cuanto a este plot, se puede observar en primer lugar que hay mucha menos gente que ha pagado una tarifa de un valor elevado, y de esas personas aproximadamente un 50% sobrevivió y por otra parte aquellos que pagaron una tarifa económica, un 40% sobrevivió, por lo tanto, si bien se observa una pequeña tendencia de que las personas con tickets más caros hayan sobrevivido, para este caso no la consideramos indispensable para la predicción que se desea realizar.

Luego, la relacion entre los supervivientes y el sexo se ve reflejada en el siguiente grafico:



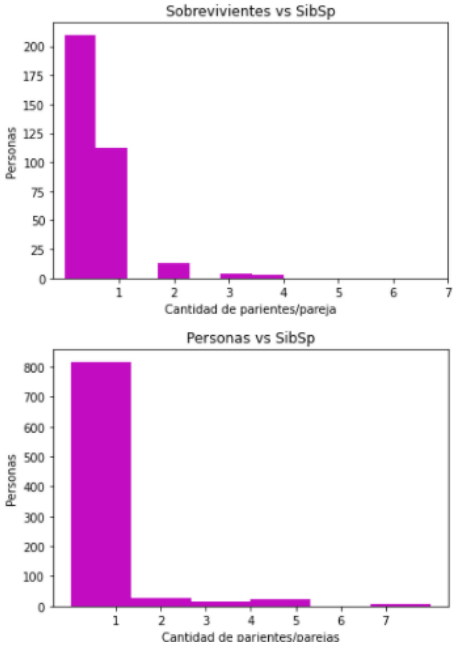
En este caso, es notable la relación que se observa, a bordo se encontraban aproximadamente 300 mujeres, de las cuales sobrevivieron un 90% lo cual implica una fuerte correlación entre las variables, y va a ser uno de los pilares en nuestro modelo de predicción.

Por otra parte, en cuanto a la relación con la clase en la que viajaban:



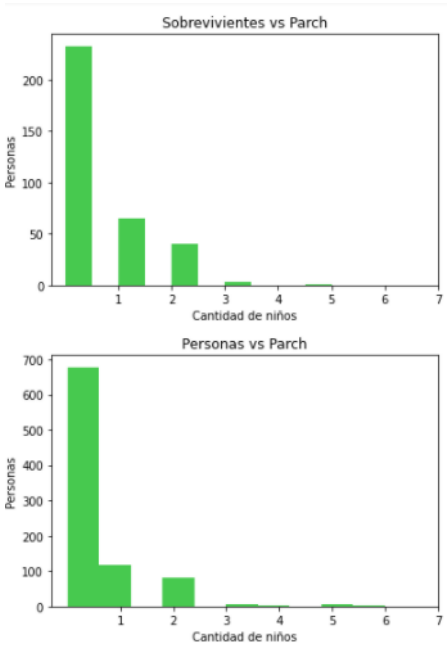
Nuevamente se observa una gran correlación entre la tasa de supervivencia y la clase en la cual los tripulantes viajaban, de unas 200 personas que iban en primera clase, 130 sobrevivieron lo cual implica un 65% de supervivencia, mientras que aquellos que viajaron en tercera clase, registran una tasa del 24%.

Por otra parte, en cuanto a la cantidad parientes/parejas a bordo



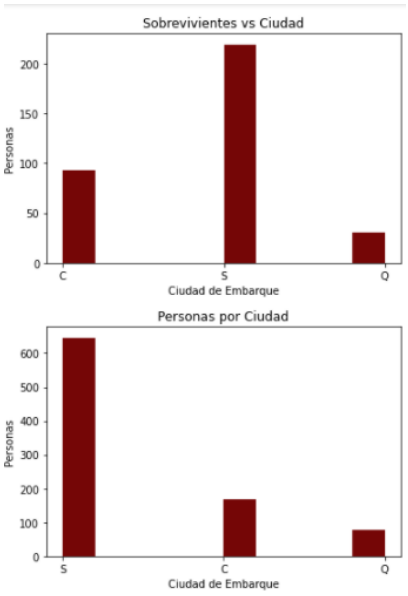
De este gráfico, se puede observar que la probabilidad de supervivencia decae muchísimo cuanto mayor es el número de parientes/parejas, pero no hay una correlación tan fuerte como para tenerla en cuenta en nuestro modelo.

Posteriormente, la relación con la cantidad de niños es la siguiente:



De nuevo, se observa una relación similar a la anterior, la mayor cantidad de supervivientes registrados no tenía niños, y a mayor cantidad de estos últimos menos probable es que sobreviva el tripulante.

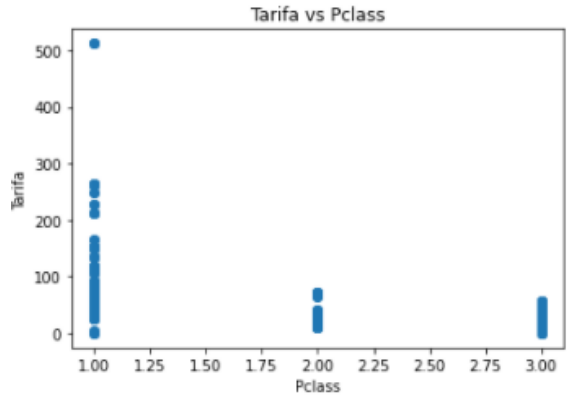
Por último, en cuanto al puerto de embarque:



En este último caso, no se observan grandes relaciones, si bien aquellas personas embarcadas en Cherbourg tienen una leve tasa de supervivencia superior, no se considera a esta variable como influyente a la hora de predecir el modelo.

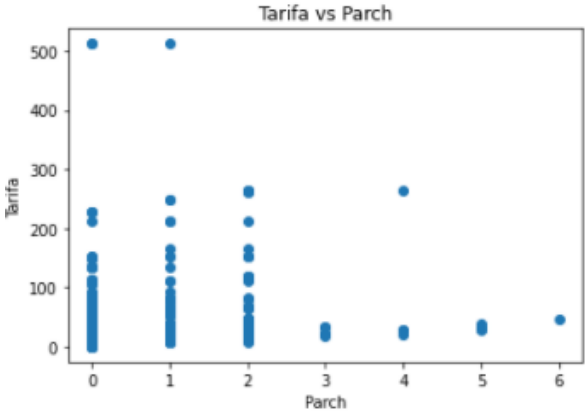
Luego, se han realizado gráficas buscando cierta relación entre las variables que expliquen al modelo, es decir, sin tener en cuenta a la columna de si sobreviven o no, de esta manera nos otorga un mejor panorama y entendimiento de la situación y del dataset.

En primer lugar, en cuanto a la relación del Fare con el resto, se obtuvieron los siguientes gráficos influyentes y representativos.

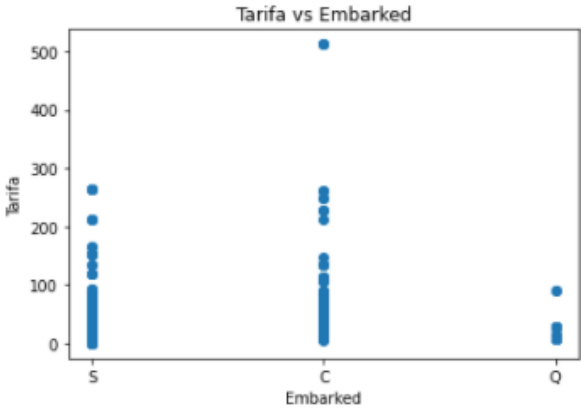


Como se puede se ver, era de esperarse que aquellos tripulantes que viajaban en primera clase, tuvieran tarifas más altas que el resto, y mediante este grafico se verifica esa presunción.

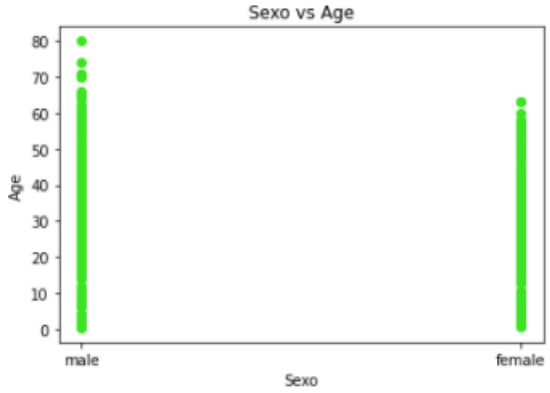
Por otra parte, en este caso se visualiza la relación que tiene con la cantidad de niños, y nuevamente como era de esperarse a mayor cantidad de niños menor iba a ser la paga de la tarifa.



Por último, se observa una marcada relación entre la tarifa y el puerto de partida, por razones que no se conocen, la vasta mayoría de las personas que han subido al Titanic desde el puerto de Queenstown, compran tickets con una tarifa muy baja en promedio a los otros 2 puertos.



En segundo lugar, entre el sexo y el resto de las variables únicamente se observó una relación significativa que era con la edad, donde las personas mayores en su mayoría eran hombres.



Luego de todos estos cálculos, se busca plantear un modelo apto de predecir si efectivamente la persona sobrevive o no, para este modelo se van a utilizar las siguientes variables: Edad, Sexo y la Pclass. En una primera medida, se va a utilizar una regresión lineal para ver que tan bien se puede ajustar el modelo a una relación lineal. Para poder realizarlo, se implementó el siguiente código, teniendo en cuenta que a la variable Sexo hubo que transformarla en una binaria para poder realizar el modelo.

```
#Luego como se puede observar se hicieron varias comparaciones y decido quedarme para predecir si sobreviven o no a l
#siguientes columnas: Edad, Pclas y Sexo, entonces me quedo con el data frame con esas 3 columnas para predecir

Dpredict = d2[["Survived","Age","Sex","Pclass"]]
Dpredict.info()

#Ahora con este data set busco armar un modelo para predicr si una persona sobrevive o no
#X serian las 3 columnas y la variable Y si sobrevive o no
Dpredict["Sex"].replace({"female":1,"male":0},inplace=True)
d2["Sex"].replace({"female":1,"male":0},inplace=True)
Dpredict.info()
X = Dpredict[["Age","Sex","Pclass"]]
Y = Dpredict[["Survived"]]
x =np.array(X)
y=np.array(Y)
X_train, X_test, y_train, y_test = train_test_split(x, y , test_size = 0.2, random_state=42, stratify= y)
reg = LinearRegression()
reg.fit(X_train,y_train)
y_pred = reg.predict(X_test)
score1 = reg.score(X_test,y_test)
print(score1)
```

El resultado del score es de 0,3481, que con nuestros conocimientos de estadística, se sabe que no es lo suficientemente bueno como para establecer una relación lineal, pero no es un valor del todo malo.

Por otra parte, se va a repetir el procedimiento pero con un dataframe distinto, en este caso se van a dropear todos los valores nulos, y antes de eso se va a dropear por completo la columna de Cabin ya que no es considerada importante y es la que más datos faltantes presenta.

El código es el siguiente:

```
df = pd.read_csv("tp.csv")
del df["Cabin"]
df = df.dropna()
print("informacion de DF")
df.info()
```

De esta forma con este nuevo dataframe se hace exactamente el mismo procedimiento con las mismas variables que en el caso anterior para luego obtener un resultado de score es de 0,3745, lo cual refleja que es levemente superior al obtenido anteriormente cunado se completaron los valores nulos con los criterios que se explicaron.

Si pasamos a la Logistic Regression con los datos de Age, Sex, Fare y Class creemos que nos dará un valor mayor para predecir los resultados.

```
logreg= LogisticRegression()
logreg.fit(X_train5,y_train5)
print(logreg.score(X_test5,y_test5))
cv_results = cross_val_score(logreg,X_train5,y_train5,cv=5)
print(cv_results)
print(np.mean(cv_results))
```

Sumado a esta regresión, añadimos un cross val score que nos dará una idea de si estamos sesgados por la elección del train. El resultado fue el siguiente.

```
Utilizando Logistic Regression
0.7972027972027972
[0.75438596 0.75438596 0.85087719 0.78070175 0.80530973]
0.7891321223412514
```

Vemos que entonces el R2 sería de 0.79 aproximadamente, un poco menor que solo con la primer iteración. Este método nos resultó, evidentemente, aceptable en términos de resultados en comparación con los anteriores propuestos y con un ajuste que consideramos bueno a priori.

Lo siguiente que tendríamos que ver es la matriz de confusión.

```
print(confusion_matrix(y_test5,y_predict))
```

```
[[74  6]
 [23 40]]
```

Vemos que esta regresión nos da muchos falsos positivos, no así falsos negativos.
El classification report nos dice que:

```
print(classification_report(y_test5,y_predict))
```

	precision	recall	f1-score	support
0	0.76	0.93	0.84	80
1	0.87	0.63	0.73	63
accuracy			0.80	143
macro avg	0.82	0.78	0.79	143
weighted avg	0.81	0.80	0.79	143

Se podría seguir con el análisis buscando otro método que nos de un mayor R2, precisión y recall.

Parte 4

En cuanto a la relación que hay entre el título de cada pasajero con la tasa de supervivencia, en primer lugar, se debe obtener el título del nombre de cada uno de los pasajeros y luego buscar graficas esos títulos con la columna de supervivencia, y mediante un value.count analizar cuantas personas con que título sobrevivieron y si a partir de eso de observa cierta relación entre ambas variables.

Para lograr separar los títulos de los nombres se utiliza el siguiente código:

```
df = pd.read_csv("tp.csv")
newdf = df["Name"].str.split(",",n=1,expand=True)
print(newdf)

newdf2 = newdf[1].str.split(".",n=1,expand=True)
print(newdf2)
newdf2.columns = ["Titulo","Nombre"]
```

De esta forma logramos obtener un nuevo dataframe con 2 columnas, una con el Titulo, y otra con el nombre de cada persona.

```
         0                                      1
0       Mr                           Owen Harris
1      Mrs   John Bradley (Florence Briggs Thayer)
2     Miss                                 Laina
3      Mrs          Jacques Heath (Lily May Peel)
4       Mr                       William Henry
..     ...                                    ...
886    Rev                              Juozas
887   Miss                     Margaret Edith
888   Miss            Catherine Helen "Carrie"
889     Mr                         Karl Howell
890     Mr                             Patrick
```

Una vez realizado esto, se aplica value.counts() para obtener la cantidad de personas que tiene cada uno de los títulos y se obtienen los siguientes resultados

```
Mr              517
Miss            182
Mrs             125
Master           40
Dr                7
Rev               6
Mlle              2
Major             2
Col               2
Capt              1
the Countess      1
Sir               1
Ms                1
Jonkheer          1
Don               1
Lady              1
Mme               1
```

Luego, se le aplica un filtro a un dataframe que contiene solamente las columnas del título y de la supervivencia, la idea es conseguir únicamente aquellas personas que hayan sobrevivido y eso se consigue de la siguiente manera.

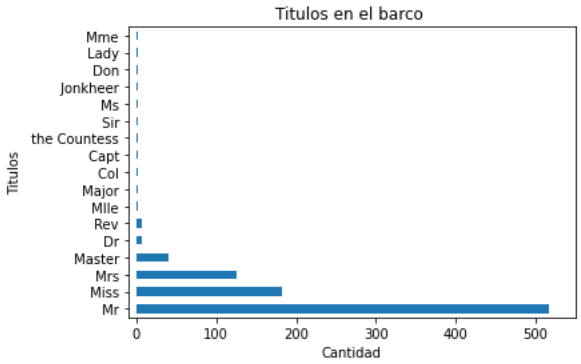
```
d2 = df[["Titulo","Survived"]]
filtro = d2["Survived"]==1
d2filtro=d2[filtro]
```

Aplicando el value.counts() quedan los siguientes valores:

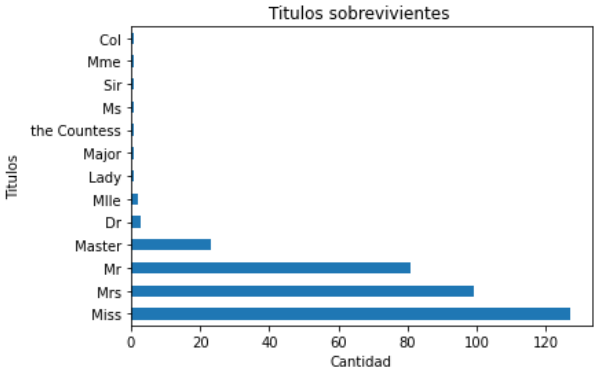
Titulo	Survived	
Miss	1	127
Mrs	1	99
Mr	1	81
Master	1	23
Dr	1	3
Mlle	1	2
the Countess	1	1
Sir	1	1
Ms	1	1
Mme	1	1
Major	1	1
Lady	1	1
Col	1	1

Ya a primera vista se observa una clara relación entre el título de Miss y la tasa de supervivencia que se reflejara a continuación en los gráficos.

Por un lado, se grafican los títulos en toda la tripulación:



Por otro lado, se grafican los títulos de aquellos que sobrevivieron:



Como se había predicho previamente hay una altísima tasa de supervivencia de aquellas personas con el título de Miss, otra relación que es posible destacar de los gráficos es de las personas que poseen el título de Master, que si bien son solo 40 a bordo, sobrevive más del 50%, lo cual implica que disponían de algún tipo de privilegio o poder a la hora de asistir a los botes salvavidas.

Por último, en cuanto a la pregunta acerca del capitán, hay varias formas de hacerlo, como bien ya se mostró previamente, aparecía en el value count de la tripulación pero no al aplicar el filtro,

por lo que se demuestra que el capitán no sobrevivo, pero si deseáramos buscar el valor únicamente solo por el hecho de ser capitán, se utiliza el siguiente código.

```
df = pd.read_csv("tp.csv")
newdf = df["Name"].str.split(",",n=1,expand=True)
print(newdf)

newdf2 = newdf[1].str.split(".",n=1,expand=True)
print(newdf2)
newdf2.columns = ["Titulo","Nombre"]
print("Aca esta con los labels bien puestos")
print(newdf2)
print(newdf2["Titulo"]==" Capt")
print("Aca encontre el Index de donde esta el capital")
print(newdf2.loc[newdf2["Titulo"] == " Capt"].index[0])
newdf2.dtypes
newdf2.describe()
print(df)
print("Aca nos dice si sobrevive o no el Capitan")
print(df.iloc[[745],[1]])
```

De esto, el paso a paso fue el siguiente, primero como se mencionó anteriormente, se separa el título del nombre del pasajero, luego a toda la columna de título se busca la posición en donde se encuentra el Capitán. Una vez obtenido el índex, se busca mediante el iloc, el valor de se índex correspondiente a la columna Survived, de esta forma al hacer esto el output es el siguiente.

```
Aca nos dice si sobrevive o no el Capitan
     Survived
745         0
```

De esta forma, se observa que en la columna de Survived aparece un 0, lo cual nos indica que efectivamente el Capitán no sobrevivió.