Comprensión de los Datos

In [1]:
```
#importa librerías
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Descripción de Variables

Pclass Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd): Categórica Nominal survival
Survival (0 = No; 1 = Yes)
name Name
sex Sex
age Age
sibsp Number of Siblings/Spouses Aboard
parch Number of Parents/Children Aboard
ticket Ticket Number
fare Passenger Fare (British pound)
cabin Cabin
embarked Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)
boat Lifeboat
body Body Identification Number
home.dest Home/Destination

Ejemplo: Crear un objeto DataFrame con base en un archivo .csv

In [5]:
```
#lee archivo csv
df = sns.load_dataset('titanic')
```

In [6]:
```
#Usa función shape para revisar el total de renglones y columnas
df.shape
```

Out[6]:
```
(891, 15)
```

In [ ]:
```
#Revisa los primeros 5 renglones del dataset usando la función head
df.head()
```

In [ ]:
```
#Revisa los últimos 5 renglones del dataset usando la función tail(
df.tail()
```

In [ ]:
```
#Revisa la información mas completa del conjunto de datos usando la
#Muestra el total de datos, las columnas y su tipo correspondiente,
```

```
df.info()
```

In [ ]:
```
#revisa cuántos valores únicos tiene cada atributo del archivo usan
df.nunique()
```

Exploración de Datos

In [ ]:
```
#utiliza la función describe() para obtener estadística básica. se
df.describe(include='all', datetime_is_numeric=True)
```

In [ ]:
```
#Revisa Valores nulos con funcion isnull().sum()
df.isnull().sum()
```

In [16]:
```
#Revisar valores únicos por columna usando función unique(): nombre
df['sex'].unique()
```

Out[16]:
```
array(['male', 'female'], dtype=object)
```

Variables Cuantitativas

Medidas de tendencia central

In [ ]:
```
#Edad
#Se puede obtener la media, mediana y moda para
mean_age = titanic['Age'].mean()
median_age =titanic['Age'].median()
mode_age = titanic['Age'].mode()
print("Mean_age:",mean_age)
print("Median_age:",median_age)
print("Mode_age:",mode_age)
```

```
Mean_age: 29.69911764705882
Median_age: 28.0
Mode_age: 0    24.0
dtype: float64
```

Conclusiones: La edad promedio fue 29 La edad al centro es 28 La edad más repetida fue de 24

Variables Categóricas

In [ ]:
```
#Para conteo  de cada valor en una columna, en orden descendente us
# nombreDataframe.columna.value_counts()
# nombreDataframe['columna'].value_counts()
df['sex'].value_counts()
```

In [10]:
```
#Revisa conteo de varias columnas
df['class'].value_counts()
```

Out[10]:

class	count
Third	491
First	216
Second	184

dtype: int64

In [ ]:
```
# Crear variable familySize que incluya la suma de las columnas Sib
# Mostrar el total por cada tamaño de familia
df['familySize'] = df['sibsp'] + df['parch']
df['familySize'].value_counts()
```

Consulta

In [11]:
```
# df.iloc[i]: Accede a la fila en la posición i.
# Acceder a la primera fila
df.iloc[0]
```

Out[11]:

	0
survived	0
pclass	3
sex	male
age	22.0
sibsp	1
parch	0
fare	7.25
embarked	S
class	Third
who	man
adult_male	True
deck	NaN
embark_town	Southampton
alive	no
alone	False

dtype: object

In [ ]:
```
# Acceder a las dos primeras filas
df.iloc[:2]
```

In [ ]:
```
#Seleccionar columnas, indicando entre corchetes [nombreColumna, no
df[['sex', 'age', 'fare']].head()
```

In [ ]:
```
#Selección de filas [indicar dataframe[columna] operador valor]
df[df['fare'] > 50]
```

In [ ]:
```
#ordenar usando funcion sort_values(by=atributo, ascending=True/fal
df.sort_values(by='fare', ascending=False).head()
```

In [ ]:
```
#Agrupar por un atributo y calcular función de agregación utilizand
df.groupby('class')['fare'].mean()
```

Crea un subconjunto de **titanic** para el costo mayor a 500

In [ ]:
```
# usa el criterio para extraer solo los boletos caros con fare > 50
df[df['fare'] > 500]
```