Comprensión de los Datos

In [1]:
```
#importa librerías
import pandas as pd
```

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 [2]:
```
#lee archivo csv
df = pd.read_csv("titanic.csv")
```

In [3]:
```
#Usa función shape para revisar el total de renglones y columnas
df.shape
```

Out[3]:
```
(891, 12)
```

In [4]:
```
#Revisa los primeros 5 renglones del dataset usando la función head()
df.head (2)
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28

In [5]:

```
#Revisa los últimos 5 renglones del dataset usando la función tail()
df.tail(6)
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	2
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	1
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	3
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	2
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	3
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	

In [7]:

```
#Revisa la información mas completa del conjunto de datos usando la función
#Muestra el total de datos, las columnas y su tipo correspondiente, dice si
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
```

In [8]:
```
#revisa cuántos valores únicos tiene cada atributo del archivo usando la fur
df.nunique()
```

Out[8]:
```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

Exploración de Datos

In [9]:
```
#utiliza la función describe() para obtener estadística básica. se puede inc
df.describe()
```

Out[9]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	89
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	3
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	4
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	1
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	3
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	51

In [13]:

```
df.describe(include='object')
```

Out[13]:

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Dooley, Mr. Patrick	male	347082	G6	S
freq	1	577	7	4	644

In [10]:

```
#Revisa Valores nulos con funcion isnull().sum()
df.isnull().sum()
```

Out[10]:

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

In [15]:

```
#Revisar valores únicos por columna usando función unique(): nombre-columna.
df.Pclass.unique()
```

Out[15]:

```
array([3, 1, 2])
```

In [16]:

```
df.Sex.unique()
```

Out[16]:

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

Variables Cuantitativas

Medidas de tendencia central

In [17]:

```
#Edad
#Se puede obtener la media, mediana y moda para
mean_age = df['Age'].mean()
median_age =df['Age'].median()
mode_age = df['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
Name: Age, dtype: float64
```

Conclusiones:

La edad promedio fue 29

La edad al centro es 28

La edad más repetida fue de 24

Variables Categóricas

In [23]:

```
#Para conteo  de cada valor en una columna, en orden descendente usar funció
# nombreDataframe.columna.value_counts()
# nombreDataframe['columna'].value_counts()
df.Sex.value_counts()
```

Out[23]:

```
Sex
male      577
female    314
Name: count, dtype: int64
```

In [9]:

```
#Revisa conteo de varias columnas
df[['Survived', 'Pclass', 'Sex', 'Embarked']].value_counts()
```

```
Out[9]: Survived  Pclass  Sex     Embarked
        0         3       male    S           231
                  2       male    S            82
        1         2       female  S            61
        0         3       female  S            55
                  1       male    S            51
        1         1       female  S            46
                                  C            42
        0         3       male    Q            36
        1         3       male    S            34
        0         3       male    C            33
        1         3       female  S            33
                  1       male    S            28
        0         1       male    C            25
        1         3       female  Q            24
                  1       male    C            17
                  3       female  C            15
                  2       male    S            15
                  3       male    C            10
        0         3       female  Q             9
                  2       male    C             8
                  3       female  C             8
        1         2       female  C             7
        0         2       female  S             6
        1         3       male    Q             3
                  2       female  Q             2
                          male    C             2
        0         1       female  S             2
                  2       male    Q             1
                  1       male    Q             1
                          female  C             1
        1         1       female  Q             1
        Name: count, dtype: int64
```

```
In [24]: df['Sex'].value_counts()
```

```
Out[24]: Sex
         male      577
         female    314
         Name: count, dtype: int64
```

```
In [10]: # Crear variable familySize que incluya la suma de las columnas SibSp y Parc
         # Mostrar el total por cada tamaño de familia
         df['familySize'] = df['SibSp'] +df['Parch']
```

```
In [11]: df
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450
...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376

891 rows × 13 columns

Consulta

```
In [12]: # df.iloc[i]: Accede a la fila en la posición i.
         # Acceder a la primera fila
         df.iloc[0]
```

```
Out[12]: PassengerId                      1
         Survived                         0
         Pclass                           3
         Name           Braund, Mr. Owen Harris
         Sex                           male
         Age                           22.0
         SibSp                            1
         Parch                            0
         Ticket                   A/5 21171
         Fare                          7.25
         Cabin                          NaN
         Embarked                         S
         familySize                       1
         Name: 0, dtype: object
```

```
In [13]: # Acceder a las dos primeras filas
         df.iloc[2]
```

```
Out[13]: PassengerId                     3
         Survived                        1
         Pclass                          3
         Name           Heikkinen, Miss. Laina
         Sex                        female
         Age                          26.0
         SibSp                           0
         Parch                           0
         Ticket           STON/O2. 3101282
         Fare                        7.925
         Cabin                         NaN
         Embarked                        S
         familySize                      0
         Name: 2, dtype: object
```

```
In [14]: #Seleccionar columnas, indicando entre corchetes [nombreColumna, nombreColum
         df[['Name', 'Age', 'Sex']]
```

Out[14]:

	Name	Age	Sex
0	Braund, Mr. Owen Harris	22.0	male
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	female
2	Heikkinen, Miss. Laina	26.0	female
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	female
4	Allen, Mr. William Henry	35.0	male
...	...	...	...
886	Montvila, Rev. Juozas	27.0	male
887	Graham, Miss. Margaret Edith	19.0	female
888	Johnston, Miss. Catherine Helen "Carrie"	NaN	female
889	Behr, Mr. Karl Howell	26.0	male
890	Dooley, Mr. Patrick	32.0	male

891 rows × 3 columns

In [15]:

```
#Selección de filas [indicar dataframe[columna] operador valor]
df[df['Age'] > 60]
```

Out[15]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
33	34	0	2	Wheadon, Mr. Edward H	male	66.0	0	0	C.A 24579
54	55	0	1	Ostby, Mr. Engelhart Cornelius	male	65.0	0	1	113509
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369
170	171	0	1	Van der hoef, Mr. Wyckoff	male	61.0	0	0	111240
252	253	0	1	Stead, Mr. William Thomas	male	62.0	0	0	113514
275	276	1	1	Andrews, Miss. Kornelia Theodosia	female	63.0	1	0	13502
280	281	0	3	Duane, Mr. Frank	male	65.0	0	0	336439
326	327	0	3	Nysveen, Mr. Johan Hansen	male	61.0	0	0	345364
438	439	0	1	Fortune, Mr. Mark	male	64.0	1	4	19950
456	457	0	1	Millet, Mr. Francis Davis	male	65.0	0	0	13509
483	484	1	3	Turkula, Mrs. (Hedwig)	female	63.0	0	0	4134
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609
545	546	0	1	Nicholson, Mr. Arthur Ernest	male	64.0	0	0	693
555	556	0	1	Wright, Mr. George	male	62.0	0	0	113807
570	571	1	2	Harris, Mr. George	male	62.0	0	0	S.W./PP 752
625	626	0	1	Sutton, Mr. Frederick	male	61.0	0	0	36963

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042
672	673	0	2	Mitchell, Mr. Henry Michael	male	70.0	0	0	C.A 24580
745	746	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1	WE/P 5735
829	830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	0	0	113572
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060

In [16]:

```
#ordenar usando funcion sort_values(by=atributo, ascending=True/false)
df.sort_values(by='Age', ascending=True)
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
803	804	1	3	Thomas, Master. Assad Alexander	male	0.42	0	1	2625
755	756	1	2	Hamalainen, Master. Viljo	male	0.67	1	1	250649
644	645	1	3	Baclini, Miss. Eugenie	female	0.75	2	1	2666
469	470	1	3	Baclini, Miss. Helene Barbara	female	0.75	2	1	2666
78	79	1	2	Caldwell, Master. Alden Gates	male	0.83	0	2	248738
...	...	...	...	...	...	...	...	...	...
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607

891 rows × 13 columns

In [17]:

```
#Agrupar por un atributo y calcular función de agregación utilizando groupby
df.groupby('Sex')[['Age', 'Fare']].mean()
```

Out[17]:

	Age	Fare
Sex		
female	27.915709	44.479818
male	30.726645	25.523893

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

In [19]:

```
# usa el criterio para extraer solo los boletos caros con fare > 50
boletos_caros = df[df["Fare"] > 500]
```

In [20]:

```
boletos_caros
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512