Comprensión de los Datos

Emiliano Hervert de la Cruz | A01412606 | Carrera: IDM

In [4]:
```
#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 [5]:
```
#lee archivo csv
df = pd.read_csv("titanic.csv")
```

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

Out[6]:
```
(891, 12)
```

In [7]:
```
#Revisa los primeros 5 renglones del dataset usando la función head()
df.head()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

In [8]:

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

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

In [9]:

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

Los atributos Age, Cabin y Embarked tienen valores nulos.

In [10]:

```
#revisa cuántos valores únicos tiene cada atributo del archivo usando la función nu
df.nunique()
```

Out[10]:

```
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 [11]:

```
#utiliza la función describe() para obtener estadística básica. se puede incluir -0
df.describe()
```

Out[11]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.00000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.20420
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.69342
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.00000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.91040
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.45420
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.00000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.32920

In [12]:

```
#utiliza la función describe(include='object') para obtener la cantidad total de va
df.describe(include='object')
```

Out[12]:

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Braund, Mr. Owen Harris	male	1601	G6	S
freq	1	577	7	4	644

In [13]:

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

Out[13]:

```
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 [14]:

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

Out[14]:

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

In [15]:

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

Out[15]:

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

Variables Cuantitativas

Medidas de tendencia central

In [16]:

```
#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 [17]:

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

Out[17]:

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

In [18]:

```
df['Sex'].value_counts()
```

Out[18]:

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

In [19]:

```
#Revisa conteo de varias columnas
```

In [20]:

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

In [21]:

```
df
```

Out[21]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7

891 rows × 13 columns

Consulta

In [22]:
```
# df.iloc[i]: Accede a la fila en la posición i.
# Acceder a la primera fila
df.iloc[0]
```

Out[22]:
```
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 [23]:
```
# Acceder a las dos primeras filas
df.iloc[:2]
```

Out[23]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	

In [24]:
```
#Seleccionar columnas, indicando entre corchetes [nombreColumna, nombreColumna]
df[['Name','Age']]
```

Out[24]:

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

891 rows × 2 columns

In [25]:

```
#Selección de filas [indicar dataframe[columna] operador valor]
sobrevivientes = df[df['Survived'] == 0]
```

In [26]:

```
sobrevivientes
```

Out[26]:

	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	7.2
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0
...	...	...	...	...	...	...	...	...	...	
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7

549 rows × 13 columns

In [27]:

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

Out[27]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060	7.7
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609	49.5
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369	7.7
745	746	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1	WE/P 5735	71.0
...	...	...	...	...	...	...	...	...	...	
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7.2
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4

549 rows × 13 columns

In [28]:

```
#Agrupar por un atributo y calcular función de agregación utilizando groupby(atribu
sobrevivientes.groupby('Pclass')['Fare'].mean()
```

Out[28]:

```
Pclass
1    64.684007
2    19.412328
3    13.669364
Name: Fare, dtype: float64
```

Crea un subconjunto de **titanic** para el costo mayor a 50

In [33]:

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

In [34]:

```
boletos_caros
```

Out[34]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	female	NaN	1	0	PC 17569	146.5
...	...	...	...	...	...	...	...	...	...	
856	857	1	1	Wick, Mrs. George Dennick (Mary Hitchcock)	female	45.0	1	1	36928	164.8
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5
867	868	0	1	Roebling, Mr. Washington Augustus II	male	31.0	0	0	PC 17590	50.4
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5
879	880	1	1	Potter, Mrs. Thomas Jr (Lily	female	56.0	0	1	11767	83.1

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
				Alexenia Wilson)						

160 rows × 13 columns