



# **INSERTAR EL REPORTE DEL TITANIC CON IMÁGENES EN UN ARCHIVO EXCEL**

**INSTITUTO POLITÉCNICO NACIONAL**

**CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN**

**JULIO 2023**

Profesor Asignado: Alan Badillo Salas

*Elaboró:* *Miguel Angel Galindo Torres*

*Correo:* *angel.galindo.torres@hotmail.com*

# **Introducción**

En esta práctica, se refuerza lo aprendido en el curso de Programación Python en el Ámbito Científico, se utiliza el archivo titanic.csv, con ayuda de la librería "Pandas" se obtiene el DataFrame del archivo anterior, se realizan una serie de modificaciones a las columnas de este DataFrame (cambio de nombres) y después una limpieza de datos, se generan una serie de cálculos al filtrar información de algunas columnas y de esta información filtrada, se obtienen los valores únicos para la columna RangoEdad (columna creada) y las cantidades de personas que caen dentro de los rangos de esta columna, de esta manera, utilizando la librería "Matplotlib", se grafica la tasa de supervivencia por rango de edades ( <= 10 años, 11 a 20 años, entre otras). El DataFrame y la gráfica (gráfica de dona), se guardan en un archivo excel, para esto se utiliza la librería "Openpyxl"

## **Justificación**

Con el fin de poner en acción los conceptos teóricos y prácticos adquiridos hasta el momento en este curso, es que se lleva a cabo esta práctica, se busca reforzar lo aprendido en la carga de DataFrames, la manipulación de estos, la creación de archivos y su modificación y la generación de gráficos.

# Pasos a resolver

Para este práctica hay una serie de requisitos a cumplir, los requisitos son los siguientes:

- Carga los datos del Titanic desde el CSV y retenlos en un DataFrame
- Genera una gráfica de pastel o barras con datos relevantes como la tasa supervivencia por edad
- Guarda los datos del DataFrame en una hoja de un archivo de Excel
- Guarda la gráfica generada en una celda de la misma hoja del archivo de Excel

Para cumplir con los requisitos de esta práctica, se necesita trabajar con un DataFrame, el cuál se puede descargar del siguiente enlace:

<https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv>

El siguiente paso es cargar el contenido del archivo csv a un DataFrame de Pandas, como se muestra en la siguiente imagen.

The screenshot shows a Jupyter Notebook cell with the following code and output:

```
#1. ADQUISICIÓN DE DATOS
#Importamos las librerías pandas y numpy
import pandas as pd
import numpy as np

#Leemos la información del archivo titanic.csv
#Obtenemos una muestra de 5 registros
info = pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv")
info.sample(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
628	629	0	3	Bastandyeff, Mr. Guentcho	male	26.0	0	0	349224	7.8958	NaN	S
166	167	1	1	Chibnall, Mrs. (Edith Martha Bowerman)	female	NaN	0	1	113505	55.0000	E33	S
381	382	1	3	Nakid, Miss. Maria ("Mary")	female	1.0	0	2	2653	15.7417	NaN	C
82	83	1	3	McDermott, Miss. Bridget Delia	female	NaN	0	0	330932	7.7875	NaN	Q
729	730	0	3	Ilmakangas, Miss. Pieta Sofia	female	25.0	1	0	STON/O2. 3101271	7.9250	NaN	S

Una vez que tenemos cargada la información en un DataFrame, podemos empezar a trabajar con ella, pero antes modificamos el nombre de las columnas para que estas sean más fáciles de comprender.

#2.- Modificando los nombres de las columnas
info.columns = ["ID", "SOBREVIVE", "CLASE", "NOMBRE", "SEXO", "EDAD", "HERMANOS", "PADRES", "TICKET", "TARIFA", "CABINA", "MUELLE"]
#Obtenemos una muestra de 5 registros
info.sample(5)
ID SOBREVIVE CLASE NOMBRE SEXO EDAD HERMANOS PADRES TICKET TARIFA CABINA MUELLE
466 467 0 2 Campbell, Mr. William male NaN 0 0 239853 0.000 NaN S
612 613 1 3 Murphy, Miss Margaret Jane female NaN 1 0 367230 15.500 NaN Q
439 440 0 2 Kvillner, Mr. Johan Henrik Johannesson male 31.0 0 0 C.A. 18723 10.500 NaN S
328 329 1 3 Goldsmith, Mrs. Frank John (Emily Alice Brown) female 31.0 1 1 363291 20.525 NaN S
715 716 0 3 Soholt, Mr. Peter Andreas Lauritz Andersen male 19.0 0 0 348124 7.650 F G73 S

A continuación, se lleva a cabo la limpieza de datos, en este caso en particular, consta de 3 aspectos, el primero implica cambiar los valores de la columna SEXO de female a MUJER y de male a HOMBRE

#3.- Limpieza de DATOS
#Aplicamos la columna SEXO (MUJER, HOMBRE)
info["SEXO"] = info["SEXO"].map({
"female": "MUJER",
"male": "HOMBRE"
})
#Obtenemos una muestra de 5 registros
info.sample(5)
ID SOBREVIVE CLASE NOMBRE SEXO EDAD HERMANOS PADRES TICKET TARIFA CABINA MUELLE
535 536 1 2 Hart, Miss. Eva Miriam MUJER 7.0 0 2 F.C.C. 13529 26.2500 NaN S
88 89 1 1 Fortune, Miss. Mabel Helen MUJER 23.0 3 2 19950 263.0000 C23 C25 C27 S
202 203 0 3 Johanson, Mr. Jakob Alfred HOMBRE 34.0 0 0 3101264 6.4958 NaN S
814 815 0 3 Tomlin, Mr. Ernest Portage HOMBRE 30.5 0 0 364499 8.0500 NaN S
47 48 1 3 O'Driscoll, Miss. Bridget MUJER NaN 0 0 14311 7.7500 NaN Q

Después, se modifican algunos valores de la columna EDAD, ya que existen registros que no cuentan con este dato, para calcular estas edades, se utiliza una función que genera números aleatorios.

# Modificamos columna EDAD ya que existen valores Nan
def generalEdades(edad):
if np.isnan(edad):
return np.random.normal(info["EDAD"].mean(),info["EDAD"].std())
else:
return edad
info["EDAD"] = info["EDAD"].map(generalEdades).map(int)
#Obtenemos una muestra de 5 registros
info.sample(5)
ID SOBREVIVE CLASE NOMBRE SEXO EDAD HERMANOS PADRES TICKET TARIFA CABINA MUELLE
202 203 0 3 Johanson, Mr. Jakob Alfred HOMBRE 34 0 0 3101264 6.4958 NaN S
713 714 0 3 Larsson, Mr. August Viktor HOMBRE 29 0 0 7545 9.4833 NaN S
487 488 0 1 Kent, Mr. Edward Austin HOMBRE 58 0 0 11771 29.7000 B37 C
42 43 0 3 Kraeff, Mr. Theodor HOMBRE 11 0 0 349253 7.8958 NaN C
676 677 0 3 Sawyer, Mr. Frederick Charles HOMBRE 24 0 0 342826 8.0500 NaN S

En seguida, se modifican algunos datos de la columna CABINA, ya que contiene valores NAN

# Modificamos columna CABINA ya que existen valores Nan
info["CABINA"] = info["CABINA"].fillna("X")
#Obtenemos una muestra de 5 registros
info.sample(5)
ID SOBREVIVE CLASE NOMBRE SEXO EDAD HERMANOS PADRES TICKET TARIFA CABINA MUELLE
386 387 0 3 Goodwin, Master. Sidney Leonard HOMBRE 1 5 2 CA 2144 46.900 X S
787 788 0 3 Rice, Master. George Hugh HOMBRE 8 4 1 382652 29.125 X Q
214 215 0 3 Kiernan, Mr. Philip HOMBRE 36 1 0 367229 7.750 X Q
839 840 1 1 Marechal, Mr. Pierre HOMBRE 32 0 0 11774 29.700 C47 C
684 685 0 2 Brown, Mr. Thomas William Solomon HOMBRE 60 1 1 29750 39.000 X S

Hasta este punto, ya que se han cargado y limpiado los datos del Titanic en un csv, se retiene la información en un DataFrame.

#Una vez modificadas las columnas y ejecutada la limpieza de datos se muestran algunos registros del DataFrame info.sample(5)												
ID	SOBREVIVIÓ	CLASE	NOMBRE	SEXO	EDAD	HERMANOS	PADRES	TICKET	TARIFA	CABINA	MUELLE	
791	792	0	2	Gaskell, Mr. Alfred	HOMBRE	16	0	0	239865	26.0000	X	S
44	45	1	3	Devaney, Miss. Margaret Della	MUJER	19	0	0	330958	7.8792	X	Q
399	400	1	2	Trout, Mrs. William H (Jessie L)	MUJER	28	0	0	240929	12.6500	X	S
812	813	0	2	Slemen, Mr. Richard James	HOMBRE	35	0	0	28206	10.5000	X	S
860	861	0	3	Hansen, Mr. Claus Peter	HOMBRE	41	2	0	350026	14.1083	X	S

A continuación, se exporta la información del DataFrame a un archivo excel, esto se lleva a cabo usando la librería Pandas

```
#Exportamos la información del DataFrame a un archivo Excel
info.to_excel("metricasTitanic.xlsx", sheet_name = "Informacion")
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1		ID	SOBREVIVIÓ	CLASE	NOMBRE	SEXO	EDAD	HERMANOS	PADRES	TICKET	TARIFA	CABINA	MUELLE									
2	0	1	2	3	3 Braund, M,HOMBRE	22	1	0	0	34231171	7.23 X	S										
3	1	2	1	1	1 Cumings, M,MUJER	38	1	0	0	PC 17599	71.2833 CRS	C										
4	2	3	1	1	1 Heikkinen, M,HOMBRE	26	0	0	0	3433802.	79.55 X	S										
5	1	4	1	1	1 Herdele, M,MUJER	35	1	0	1	313801	11.3123 X	S										
6	4	5	0	3	3 Allen, Mr. H,HOMBRE	35	0	0	0	313434	8.05 X	S										
7	5	6	0	3	3 Moran, M,HOMBRE	44	0	0	0	343087	8.4051 X	Q										
8	6	7	0	3	3 Weston, M,HOMBRE	54	0	0	0	343433	51.33 X	446										
9	7	8	0	3	3 Palsson, M,HOMBRE	2	3	0	0	349909	21.075 X	S										
10	8	9	1	3	3 Johnson, M,MUJER	27	0	0	0	347742	11.1313 X	S										
11	9	10	1	3	3 Homann, M,HOMBRE	24	1	0	0	347742	30.27 X	C										
12	10	11	1	3	3 Sanderson,MUJER	4	1	1	1	PP 9549	16.7 06	S										
13	11	12	1	1	1 Bonetti, M,MUJER	58	0	0	1	313781	26.50 C103	S										
14	12	13	0	3	3 Lightbody, M,HOMBRE	29	0	0	0	343433	33.51 X	S										
15	13	14	0	3	3 Anderson, M,HOMBRE	39	1	0	0	347082	31.275 X	S										
16	14	15	0	3	3 Westcott, M,MUJER	14	0	0	0	343646	7.854 X	S										
17	15	16	0	3	3 Cumings, M,HOMBRE	55	0	0	0	343652	20.50 X	S										
18	16	17	0	3	3 Rice, Mabel,HOMBRE	2	4	0	0	348252	29.125 X	Q										
19	17	18	1	2	2 Williams, M,HOMBRE	59	0	0	0	344373	13 X	S										
20	18	19	0	3	3 Wicks, M,HOMBRE	31	0	0	0	343610	20.50 X	S										
21	19	20	1	3	3 Masselman,MUJER	21	0	0	0	02449	7.225 X	C										
22	20	21	0	3	2 Verney, M,HOMBRE	35	0	0	0	349865	26 X	S										
23	21	22	1	3	3 McGowan, M,HOMBRE	24	0	0	0	349865	20.50 X	S										
24	22	23	1	3	3 McGowan, MUJER	15	0	0	0	340923	8.025 X	Q										
25	23	24	1	1	1 Stoker, Mr. H,HOMBRE	28	0	0	0	313788	35.5 AG											
26	24	25	0	3	3 Allison, M,HOMBRE	4	0	0	0	343652	20.50 X	S										
27	25	26	1	3	3 Allison, M,MUJER	38	1	0	0	347077	31.3875 X	S										
28	26	27	0	3	3 Emir, Mr. H,HOMBRE	26	0	0	0	34631	7.225 X	C										
29	27	28	0	3	3 Fyvie, M,HOMBRE	35	0	0	0	349865	20.50 X	S										

Para realizar el cálculo de la información estadística, se han construido 2 funciones. La primera función filtra el DataFrame original con respecto a algún valor específico de una sola columna, el valor devuelto de esta función puede ser la cantidad de filas que cumplen con la condición específica o puede devolver el DataFrame filtrado que cumplen de igual manera con la condición específica

```

def totalesColumna(infoOriginal, columna, operador, condicion, filas):
    if operador == '>':
        infoFiltrar = infoOriginal[columna] > condicion
    elif operador == '<':
        infoFiltrar = infoOriginal[columna] < condicion
    elif operador == '=':
        infoFiltrar = infoOriginal[columna] == condicion
    elif operador == '!=':
        infoFiltrar = infoOriginal[columna] != condicion
    elif operador == '>=':
        infoFiltrar = infoOriginal[columna] >= condicion
    elif operador == '<=':
        infoFiltrar = infoOriginal[columna] <= condicion
    else:
        infoFiltrar = infoOriginal[columna] != condicion

    infofiltrada = infoOriginal[infoFiltrar]

    if filas == True:
        return len(infofiltrada)
    else:
        return infofiltrada

```

La segunda función filtra el DataFrame original con respecto a algún valor específico de dos columnas, el valor devuelto de esta función puede ser la cantidad de filas que cumplen con las condiciones específica o puede devolver el DataFrame filtrado que cumplen de igual manera con las condiciones que se pasan como argumentos.

Para construir la gráfica que nos permita visualizar los sobrevivientes por rango de edades, creamos una nueva columna al DataFrame, de esta manera, podremos obtener los valores únicos que aparecerán en las etiquetas propias de este tipo de gráfica.

## Total de supervivientes

```
#Añadimos columnas RangoEdades
listaRango = []
for index, row in info.iterrows():
    if row["EDAD"] <= 10:
        listaRango.append("Menor de 10 años")
    elif row["EDAD"] >= 11 and row["EDAD"] <= 20:
        listaRango.append("Entre 11 y 20 años")
    elif row["EDAD"] >= 21 and row["EDAD"] <= 30:
        listaRango.append("Entre 21 y 30 años")
    elif row["EDAD"] >= 31 and row["EDAD"] <= 40:
        listaRango.append("Entre 31 y 40 años")
    elif row["EDAD"] >= 41 and row["EDAD"] <= 50:
        listaRango.append("Entre 41 y 50 años")
    elif row["EDAD"] >= 51 and row["EDAD"] <= 60:
        listaRango.append("Entre 51 y 60 años")
    elif row["EDAD"] >= 61 and row["EDAD"] <= 70:
        listaRango.append("Entre 61 y 70 años")
    elif row["EDAD"] >= 71 and row["EDAD"] <= 80:
        listaRango.append("Entre 71 y 80 años")
    else:
        listaRango.append("Entre 81 y 90 años")

info["RANGOEDAD"] = listaRango
```

El primer cálculo que se lleva a cabo sobre el DataFrame, es obtener información de los sobrevivientes y el número de estos. Se utiliza una misma función para obtener ambos datos, es decir, se obtiene un DataFrame filtrado, que se utiliza a lo largo de toda la práctica y se obtiene la cantidad de filas de este DataFrame para los cálculos se las tasas de los sobrevivientes.

## Total de sobreviventes

```
#sobrevida, se emplea solo una vez y sirve como parámetro de entrada para la función totales2Columnas
#El último parámetro (False) indica que el valor devuelto por la función es un DataFrame
sobrevida = totales1Columna(info, "SOBREVIVE", "=", True, False)

#totalViven, es La cantidad de registros que cumplen con las condiciones para la función totales1Columna
#El último parámetro (True) indica que el valor devuelto por la función es un valor entero
totalViven = totales1Columna(info, "SOBREVIVE", "=", True, True)
totalViven
```

342

Para el siguiente cálculo, se requiere conocer el total de sobrevivientes cuya edad sea menor o igual a 10 años.

## Sobrevivientes menores a 10 años

```
#Para saber la cantidad de personas que sobreviven y son menores de 10 años
edades0a10 = totales2Columnas(info, "EDAD", "<=", 10, "and", "SOBREVIVE", "=", True, True)
```

44

Conociendo el dato anterior, ahora podemos calcular la tasa de sobrevivientes menores o iguales a 10 años.

## Tasa de sobrevivientes menores a 10

```
#Calculando Tasas de Supervivencia del rango menor a 10 años
print("{:.2f} %".format(edades0a10 / totalViven * 100))
```

12.87 %

Calculamos el total de sobrevivientes cuyas edades comprendan entre 11 y 20 años.

#### Sobrevivientes entre 11 y 20

```
#Para saber la cantidad de personas que sobreviven entre 11 y 20 años
edades11a20 = totales2Columnas(sobrevive,"EDAD",">=",11,"and","EDAD","<=",20,True)
edades11a20
```

51

La tasa de sobrevivientes para el rango de edad entre 11 y 20 años es la siguiente:

#### Tasa de sobrevivientes entre 11 y 20

```
#Calculando Tasas de Supervivencia del rango Entre 11 y 20 años
print("{:.2f} %".format((edades11a20 / totalViven) * 100))
```

14.91 %

En este cálculo, se requiere conocer el número total de sobrevivientes entre 21 y 30 años

#### Sobrevivientes entre 21 y 30

```
#Para saber la cantidad de personas que sobreviven entre 21 y 30 años
edades21a30 = totales2Columnas(sobrevive,"EDAD",">=",21,"and","EDAD","<=",30,True)
edades21a30
```

97

La tasa de sobrevivientes para el rango de edad anterior es el siguiente:

#### Tasa de sobrevivientes entre 21 y 30

```
#Calculando Tasas de Supervivencia del rango Entre 21 y 30 años
print("{:.2f} %".format((edades21a30 / totalViven) * 100))
```

28.36 %

El número de sobrevivientes entre 31 y 40 años corresponde a lo que se muestra en la imagen:

#### Sobrevivientes entre 31 y 40

```
#Para saber la cantidad de personas que sobreviven entre 31 y 40 años
edades31a40 = totales2Columnas(sobrevive,"EDAD",">=",31,"and","EDAD","<=",40,True)
edades31a40
```

80

La tasa de sobrevivientes para el rango de edad entre 31 y 40 es:

### Tasa de sobrevivientes entre 31 y 40

```
#Calculando Tasas de Supervivencia del rango Entre 31 y 40 años
print("{:.2f} %".format((edades31a40 / totalViven) * 100))
23.39 %
```

En el siguiente cálculo, se obtiene el total de sobrevivientes de edades entre los 41 años y 50 años:

### Sobrevivientes entre 41 y 50

```
#Para saber la cantidad de personas que sobreviven entre 41 y 50 años
edades41a50 = totales2Columnas(sobrevive,"EDAD",">=",41,"and", "EDAD","<=",50,True)
edades41a50
```

44

La tasa de los sobrevivientes de edades entre los 41 años y 50 años corresponde a:

### Tasa de sobrevivientes entre 41 y 50

```
#Calculando Tasas de Supervivencia del rango Entre 41 y 50 años
print("{:.2f} %".format((edades41a50 / totalViven) * 100))
12.87 %
```

Las personas que sobrevivieron y cuyas edades están entre los 51 y 60, suman un total de:

### Sobrevivientes entre 51 y 60

```
#Para saber la cantidad de personas que sobreviven entre 51 y 60 años
edades51a60 = totales2Columnas(sobrevive,"EDAD",">=",51,"and", "EDAD","<=",60,True)
edades51a60
```

20

La tasa de las personas que sobrevivieron y pertenecen al rango anterior de edades es de:

### Tasa de sobrevivientes entre 51 y 60

```
#Calculando Tasas de Supervivencia del rango Entre 51 y 60 años
print("{:.2f} %".format((edades51a60 / totalViven) * 100))
5.85 %
```

Los sobrevivientes entre 61 y 70 años son en total:

### Sobrevivientes entre 61 y 70

```
#Para saber la cantidad de personas que sobreviven entre 61 y 70 años
edades61a70 = totales2Columnas(sobrevive,"EDAD",">=",61,"and", "EDAD","<=",70,True)
edades61a70
```

5

La tasa para los sobrevivientes entre 61 y 70 años es de:

#### Tasa de sobrevivientes entre 61 y 70

```
#Calculando Tasas de Supervivencia del rango Entre 61 y 70 años
print("{:.2f} %".format((edades61a70 / totalViven) * 100))
1.46 %
```

Los sobrevivientes en el rango de edades de entre 71 y 80 años suman en total:

#### Sobrevivientes entre 71 y 80

```
#Para saber la cantidad de personas que sobreviven entre 71 y 80 años
edades71a80 = totales2Columnas(sobrevive,"EDAD",">=",71,"and","EDAD","<=",80,True)
edades71a80
1
```

Y por último, la tasa de las personas que sobrevivieron en el rango de edad anterior es:

#### Tasa de sobrevivientes entre 71 y 80

```
#Calculando Tasas de Supervivencia del rango Entre 71 y 80 años
print("{:.2f} %".format((edades71a80 / totalViven) * 100))
0.29 %
```

Con las cantidades de supervivientes por rango de edades y las tasas de supervivencia correspondientes, creamos una gráfica para la mejor visualización e interpretación de los datos. El primer paso para la creación de la gráfica (de tipo dona) es importar la librería matplotlib y obtener las diferentes etiquetas que se han de mostrar en el gráfico

```
#Generando una gráfica de donas, se importa la Librería matplotlib
import matplotlib.pyplot as plt

#se obtienen las diferentes etiquetas para la gráfica
etiquetas = sobrevive["RANGOEDAD"].unique()
etiquetas

array(['Entre 31 y 40 años', 'Entre 21 y 30 años', 'Entre 11 y 20 años',
       'Menor de 10 años', 'Entre 51 y 60 años', 'Entre 41 y 50 años',
       'Entre 61 y 70 años', 'Entre 71 y 80 años'], dtype=object)
```

A continuación se calculan los pesos para las etiquetas que se obtuvieron en el paso anterior.

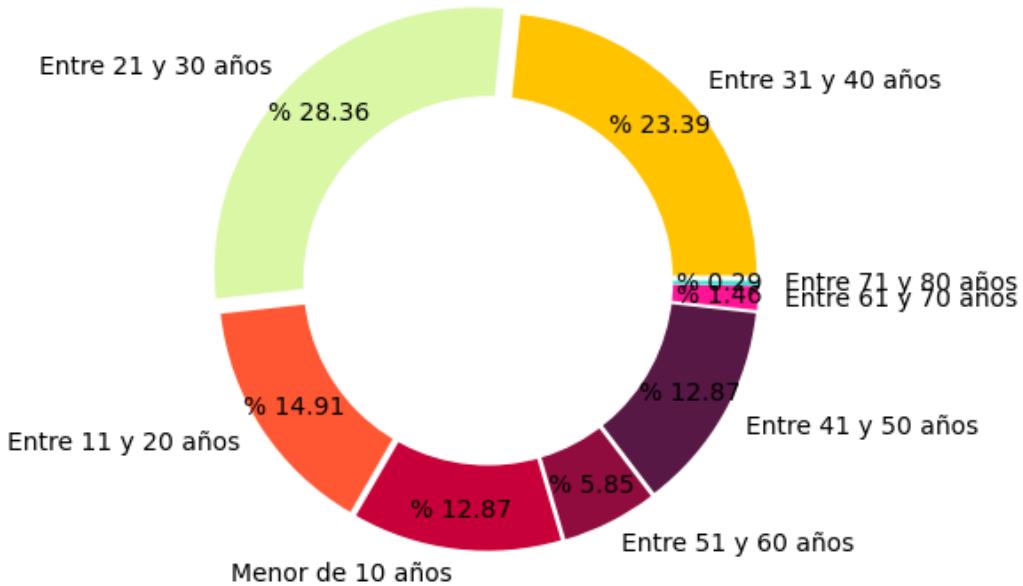
```
#Se obtienen los pesos para cada etiqueta
pesos = [sobrevive["RANGOEDAD"][sobrevive["RANGOEDAD"] == label].count() for label in etiquetas]
pesos

[80, 97, 51, 44, 20, 44, 5, 1]
```

Se definen algunas características de la gráfica como los colores, la distancia entre las etiquetas y sus pesos, el título de la gráfica, entre algunas otras, se guarda la imagen, el resultado es el siguiente.

```
#Graficando la información
colores = ["#FFC300", "#DAF7A6", "#FF5733", "#C70039", "#900C3F", "#581845", "#FF1493", "#40E0D0"]
maxPeso = max(pesos)
exp = []
for item in pesos:
    if item == maxPeso:
        exp.append(0.06)
    else:
        exp.append(0.03)
#Se especifica los pesos, etiquetas, colores, y demás parámetros
plt.pie(pesos,
         labels = etiquetas,
         autopct = "%%.2f",
         pctdistance = 0.85,
         colors = colores,
         explode = exp
        )
#Se agrega un título a la gráfica y se crea
circle = plt.Circle((0,0), 0.7, color = "white")
plt.gcf().gca().add_artist(circle)
plt.title("Tasa de supervivencia por Rango de Edades")
plt.savefig("GraficaDona")
plt.show()
```

Tasa de supervivencia por Rango de Edades



Se instala la librería openpyxl para añadir imágenes a un archivo excel.

```
! pip install openpyxl
Requirement already satisfied: openpyxl in d:\users\mgalindo\anaconda3\lib\site-packages (3.0.10)
Requirement already satisfied: et_xmlfile in d:\users\mgalindo\anaconda3\lib\site-packages (from openpyxl) (1.1.0)
```

Se abre el archivo excel, se especifica la imagen y el lugar dentro de la hoja en donde se requiera colocar la imagen, se guardan las modificaciones realizadas al archivo y el resultado se muestra a continuación.

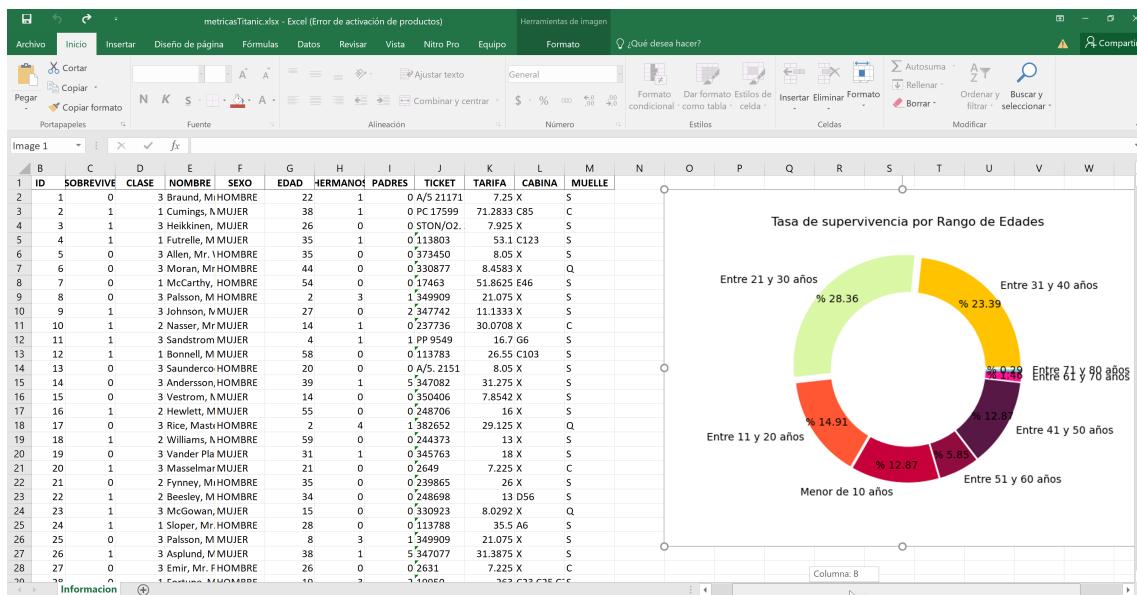
```
#Exportando la imagen de la gráfica de donas al excel que contiene la información del DataFrame

#Se hace uso de la librería openpyxl para trabajar con imágenes en archivos excel.
from openpyxl import load_workbook
from openpyxl.drawing.image import Image

#Se carga el archivo excel, creado anteriormente
libro = load_workbook("metricasTitanic.xlsx")
hoja = libro.active

#Se especifica que imagen se va a cargar en el excel y la ubicación de esta
img = Image("GraficaDonas.png")
hoja.add_image(img, "O2")

#Se guarda el excel con las modificaciones realizadas (imagen exportada)
libro.save("metricasTitanic.xlsx")
```



# Conclusiones

Debido a que se llevó a cabo una limpieza de datos que en un caso específico generó edades (número enteros) con números aleatorios, la cantidad de personas con respecto a las edades ( $\geq 18$ ,  $< 18$  ó  $\geq 50$ ), las tasas calculadas y la gráfica que se genera, puede variar de una corrida de información a otra, sin embargo, esta variación no es significante ya que la mayoría de los registros en la columna Edad del DataFrame tiene información válida. Con respecto a las funciones creadas para filtrar la información y obtener la cantidad de filas como resultados, estas se pueden unir en una sola función que lleve a cabo el trabajo de ambas, así como añadir más operadores lógicos a las mismas, ya que por el momento solo se hace uso del operador AND, faltando los operadores OR y NOT. La cantidad de gráficas generadas en esta práctica ha sido solo 1 (gráfica de donas) pero dada la información disponible y los diferentes filtros que se pueden aplicar al DataFrame, es posible generar otros tipos de gráfica (barras, puntos), Desde el punto de vista de la información cargada en el archivo csv, se puede concluir que la tasa de supervivencia para el rango de edades de entre 21 y 30 años, es la más alta, con un 23.36% mientras la más baja corresponde al rango de edades que va de los 71 a los 80 años.