

TutorialMatplotlibReportes

June 6, 2021

1 Tutorial de Matplot para graficas y de Reportes pagermill

A continuación se detalla un pequeño tutorial de como utilizar matplotlib y pagermill para la generacion de graficas y reportes respectivamente, este tutorial se basa en tres librerias: - Matplotlib - Numpy - Pandas

Al finalizar el estudiante estará en la capacidad de generar graficas y enviar parametros para la realización de reportes utilizando Notebook. Además permite la lectura de archivos .csv y de diferentes tipos de graficos.

```
[1]: #importar las librerias necesarias
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

1.1 Construccion de un grafico basico

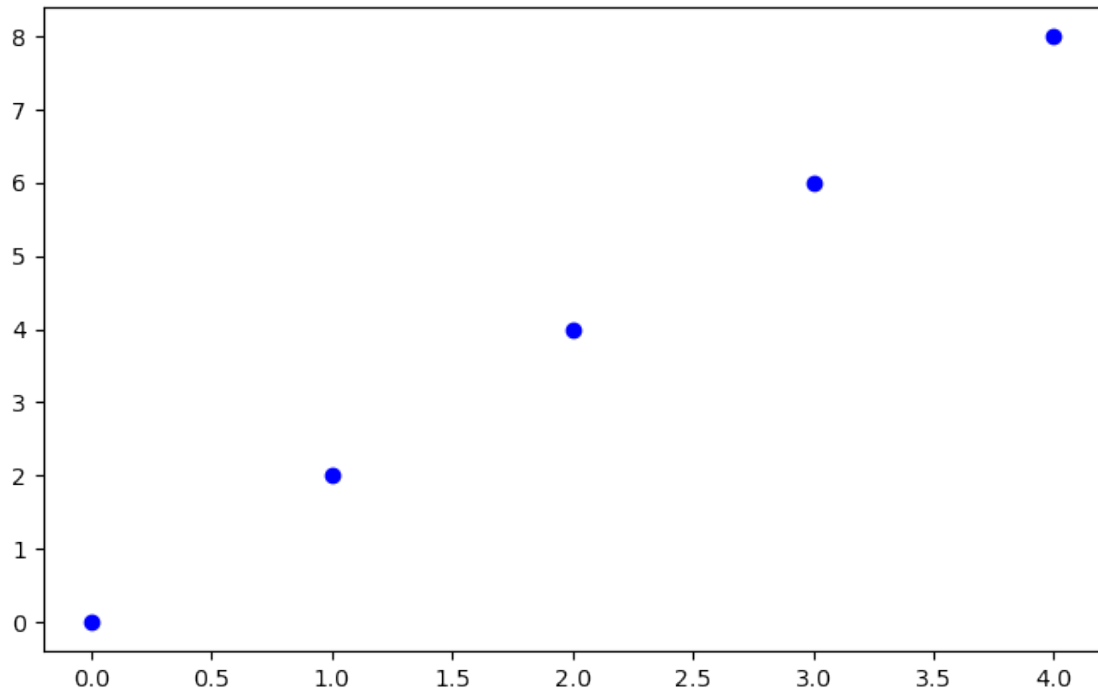
Para la contriccion de este primer grafico utilizaremos dos arreglos

```
[2]: x = [0,1,2,3,4]
y = [0,2,4,6,8]

# Asignamos un tamaño y el numero de pixeles por pulgada.
plt.figure(figsize=(8,5), dpi=100)

# Graficos en formato punto
plt.plot(x,y, 'bo')
```

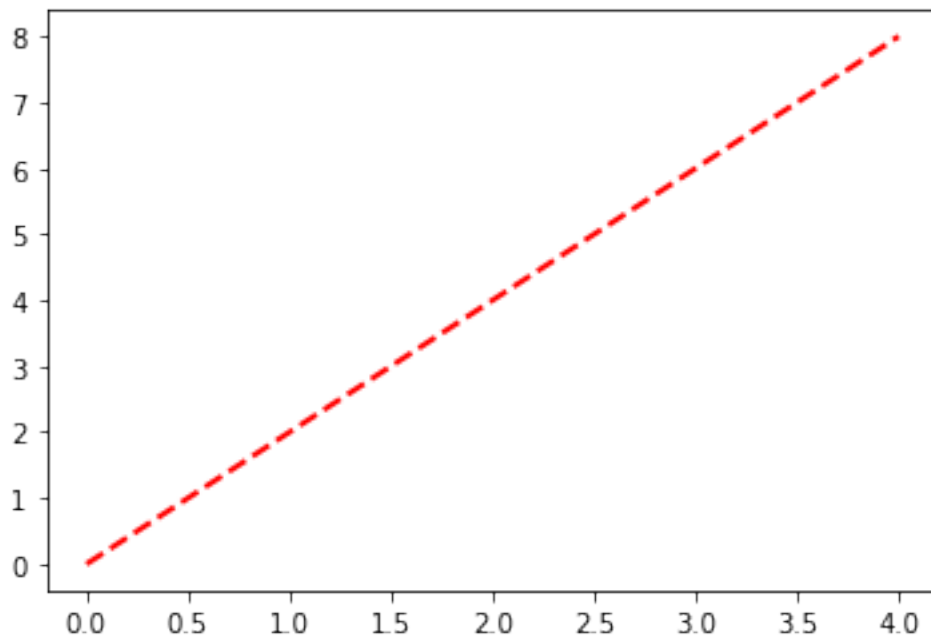
```
[2]: [<matplotlib.lines.Line2D at 0x1f08c1b2eb0>]
```



[3]: *#Para graficar en un formato linea utilizamos el siguiente comando*

#Graficamos linea

```
plt.plot(x,y, color='red', linewidth=2, linestyle='--', label='Linea')  
plt.show()
```



```
[4]: # Generar un rango de puntos para ello vamos a utilizar np.arange(inicio, fin,
    ↪ paso)

x2 = np.arange(0,4.5,0.5)
y2 = np.arange(0,4.5,0.5)

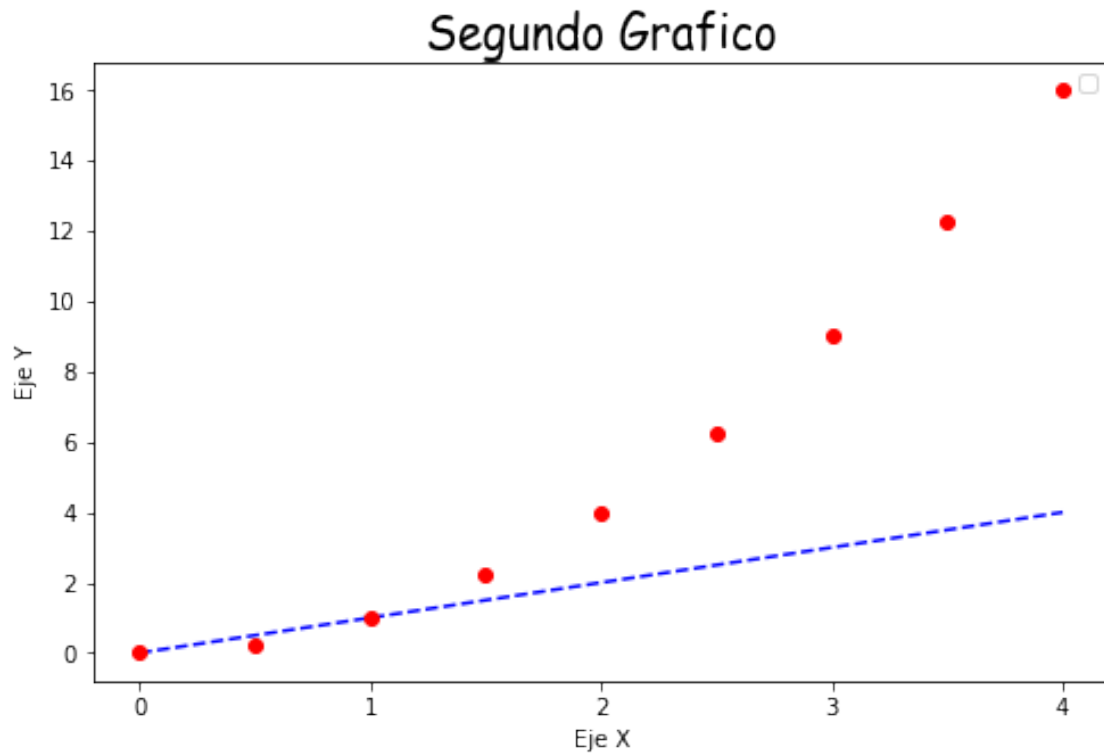
plt.figure(figsize=(8,5))
plt.title('Segundo Grafico', fontdict={'fontname':'Comic Sans MS', 'fontsize':
    ↪ 20})
plt.xlabel('Eje X')
plt.ylabel('Eje Y')

#Graficar
plt.plot(x2,y2,'b--')
#Segunda linea con puntos y elevado al cuadrado el arreglo x2
plt.plot(x2,x2**2,'ro')
# Asignar Escala de eje en X
plt.xticks([0,1,2,3,4])
#Agregar la leyenda al grafico
plt.legend()

#Guardar la imagen en un archivo
plt.savefig('segundografico.png',dpi=300)

plt.show()
```

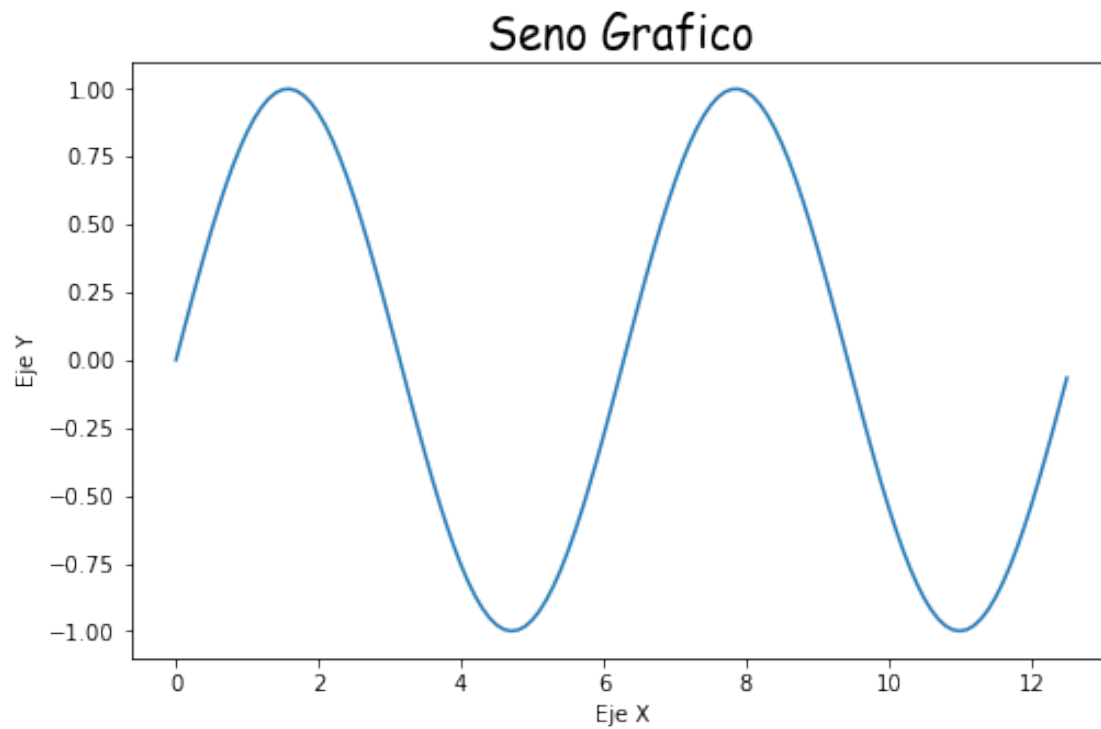
No handles with labels found to put in legend.



```
[5]: # Graficar funciones matematicas

x3 = np.arange(0, 4*np.pi, 0.1)
y3 = np.sin(x3)

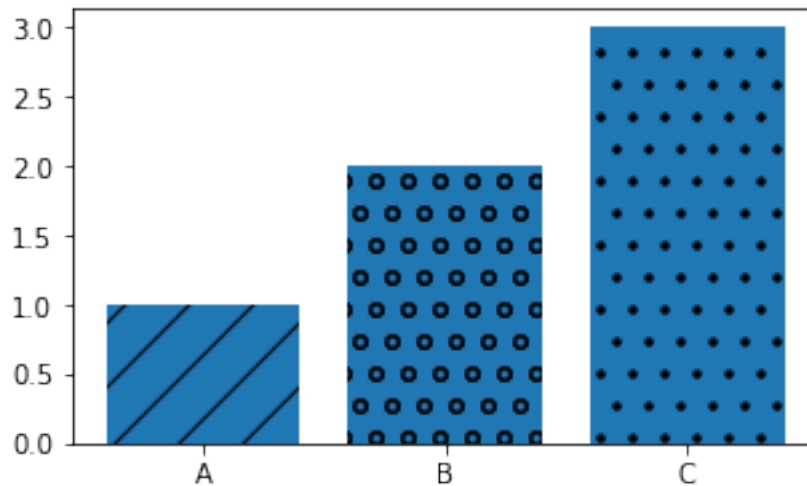
plt.figure(figsize=(8,5))
plt.title('Seno Grafico', fontdict={'fontname':'Comic Sans MS', 'fontsize': 20})
plt.xlabel('Eje X')
plt.ylabel('Eje Y')
plt.plot(x3,y3)
plt.show()
```



```
[6]: #Grafico de Barras
etiquetas = ['A', 'B', 'C']
valores = np.arange(1,4,1)
plt.figure(figsize=(5,3))
barras = plt.bar(etiquetas, valores)

#Patrones dentro de las barras
patrones = ['/', 'o', '.']
for bar in barras:
    bar.set_hatch(patrones.pop(0))

plt.show()
```



```
[7]: #Trabajar con datos en formato .csv

datos = pd.read_csv('gas_precios.csv')

plt.figure(figsize=(5,3))

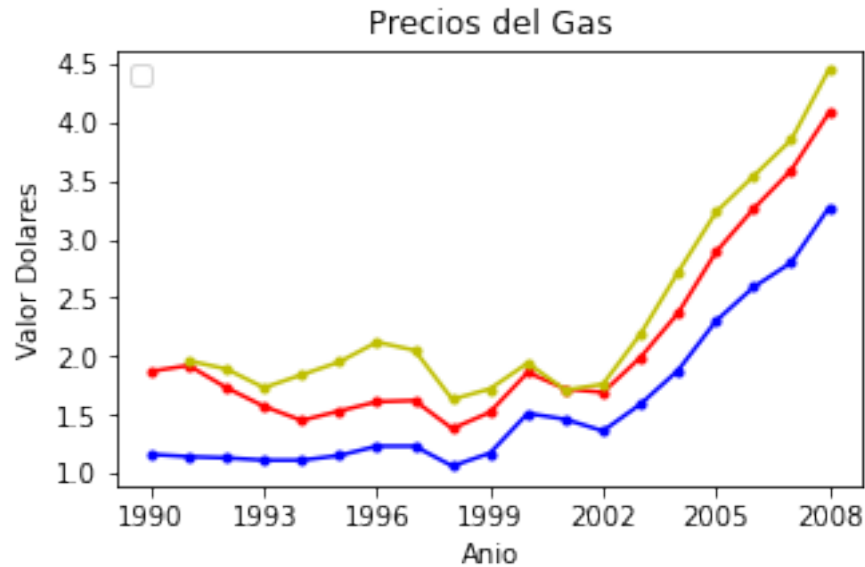
plt.title('Precios del Gas')

plt.plot(datos.Year, datos.USA, 'b.-')
plt.plot(datos.Year, datos.Canada, 'r.-')
plt.plot(datos.Year, datos.Australia, 'y.-')

plt.xticks(datos.Year[::3].tolist())

plt.xlabel('Anio')
plt.ylabel('Valor Dolares')
plt.legend()
plt.show()
```

No handles with labels found to put in legend.



```
[8]: #Trabajar con otro tipo de datos Fifa
fifa = pd.read_csv('fifa_datos.csv')
#imprimir los primeros 5 datos del archivo
fifa.head(5)
```

```
[8]: Unnamed: 0      ID      Name  Age  \
0      0  158023      L. Messi   31
1      1  20801  Cristiano Ronaldo  33
2      2  190871      Neymar Jr   26
3      3  193080      De Gea    27
4      4  192985      K. De Bruyne  27

      Photo Nationality  \
0  https://cdn.sofifa.org/players/4/19/158023.png  Argentina
1  https://cdn.sofifa.org/players/4/19/20801.png    Portugal
2  https://cdn.sofifa.org/players/4/19/190871.png    Brazil
3  https://cdn.sofifa.org/players/4/19/193080.png    Spain
4  https://cdn.sofifa.org/players/4/19/192985.png    Belgium

      Flag  Overall  Potential  \
0  https://cdn.sofifa.org/flags/52.png      94      94
1  https://cdn.sofifa.org/flags/38.png      94      94
2  https://cdn.sofifa.org/flags/54.png      92      93
3  https://cdn.sofifa.org/flags/45.png      91      93
4  https://cdn.sofifa.org/flags/7.png       91      92

      Club  ...  Composure  Marking  StandingTackle  SlidingTackle  \
```

0	FC Barcelona	...	96.0	33.0	28.0	26.0
1	Juventus	...	95.0	28.0	31.0	23.0
2	Paris Saint-Germain	...	94.0	27.0	24.0	33.0
3	Manchester United	...	68.0	15.0	21.0	13.0
4	Manchester City	...	88.0	68.0	58.0	51.0

	GK Diving	GK Handling	GK Kicking	GK Positioning	GK Reflexes	Release Clause
0	6.0	11.0	15.0	14.0	8.0	€226.5M
1	7.0	11.0	15.0	14.0	11.0	€127.1M
2	9.0	9.0	15.0	15.0	11.0	€228.1M
3	90.0	85.0	87.0	88.0	94.0	€138.6M
4	15.0	13.0	5.0	10.0	13.0	€196.4M

[5 rows x 89 columns]

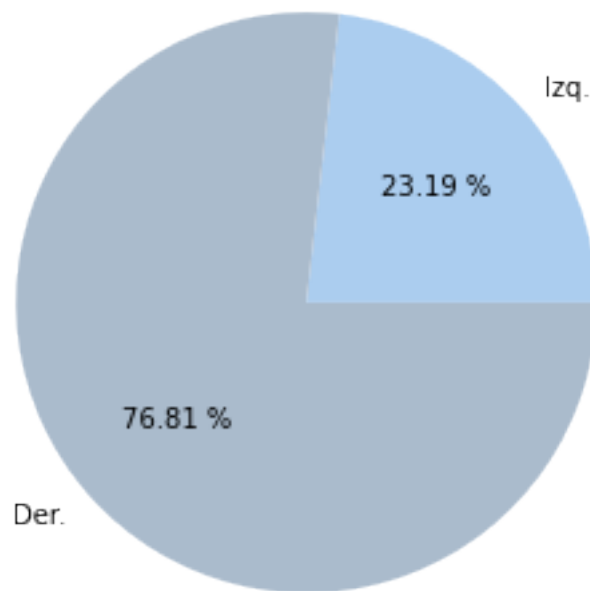
```
[9]: # Generar un grafico de cual es su pie diestro

izquierdo = fifa.loc[fifa['Preferred Foot'] == 'Left'].count()[0]
derecho = fifa.loc[fifa['Preferred Foot'] == 'Right'].count()[0]

plt.figure(figsize=(8,5))

etiquetas = ['Izq.', 'Der.']
colores = ['#abcdef', '#aabbcc']
plt.pie([izquierdo, derecho], labels=etiquetas, colors=colores, autopct='%2f_
↪%%')
plt.title('Pie de juego preferido')
plt.show()
```

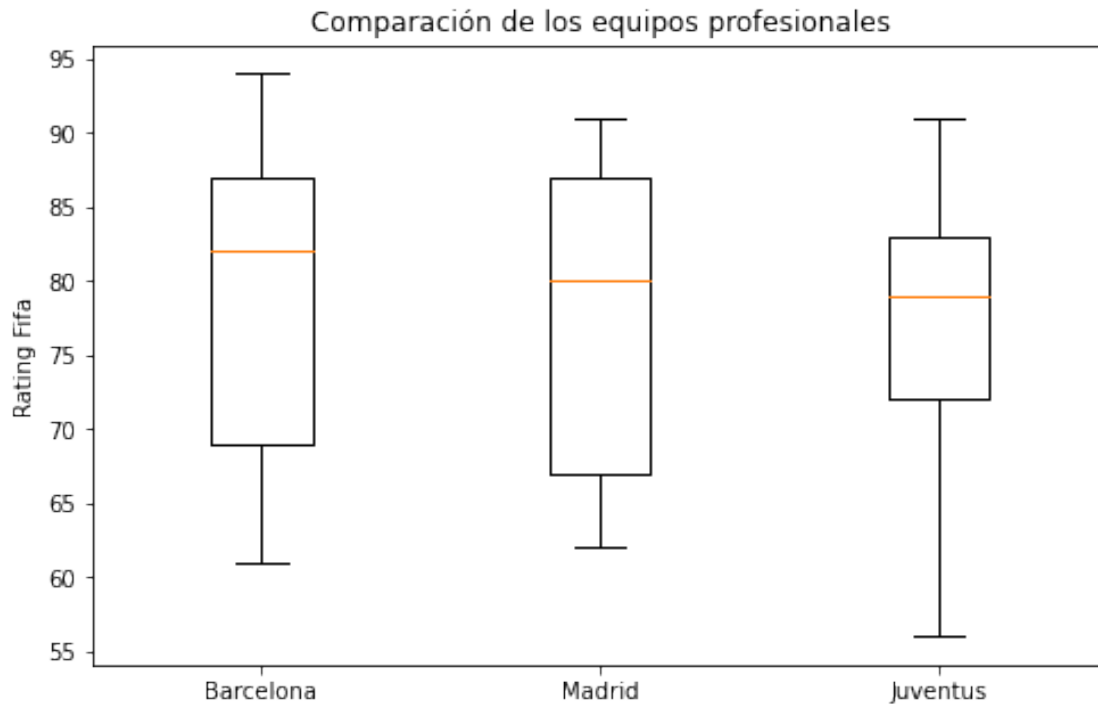

Pie de juego preferido



```
[10]: # Cuadro de barras y bigotes

plt.figure(figsize=(8,5))

barcelona = fifa.loc[fifa.Club == 'FC Barcelona']['Overall']
madrid = fifa.loc[fifa.Club == 'Real Madrid']['Overall']
juventus = fifa.loc[fifa.Club == 'Chelsea']['Overall']
bp = plt.boxplot([barcelona, madrid, juventus], labels=['Barcelona', 'Madrid', 'Juventus'])
plt.title('Comparación de los equipos profesionales')
plt.ylabel('Rating Fifa')
plt.show()
```



1.2 Tarea

1. Con los datos de Fifa, organizar a los jugadores de acuerdo al peso en las siguientes escalas y generar un cuadro tipo PIE

- Debajo 125 Lbs.
- 125-150
- 150-175
- 175 o superior

```
[11]: #print(fifa.dtypes)

#print(fifa.sort_values(by=['Weight'], ascending=[True]))

#fifa.loc["Weight"]
#fifa[['Name', 'Weight']]

#fifa['Weight'] = fifa['Weight'].astype(float)

#fifa.sort_values(by=['Weight'], ascending=[True])

rango1 = 0
rango2 = 0
```

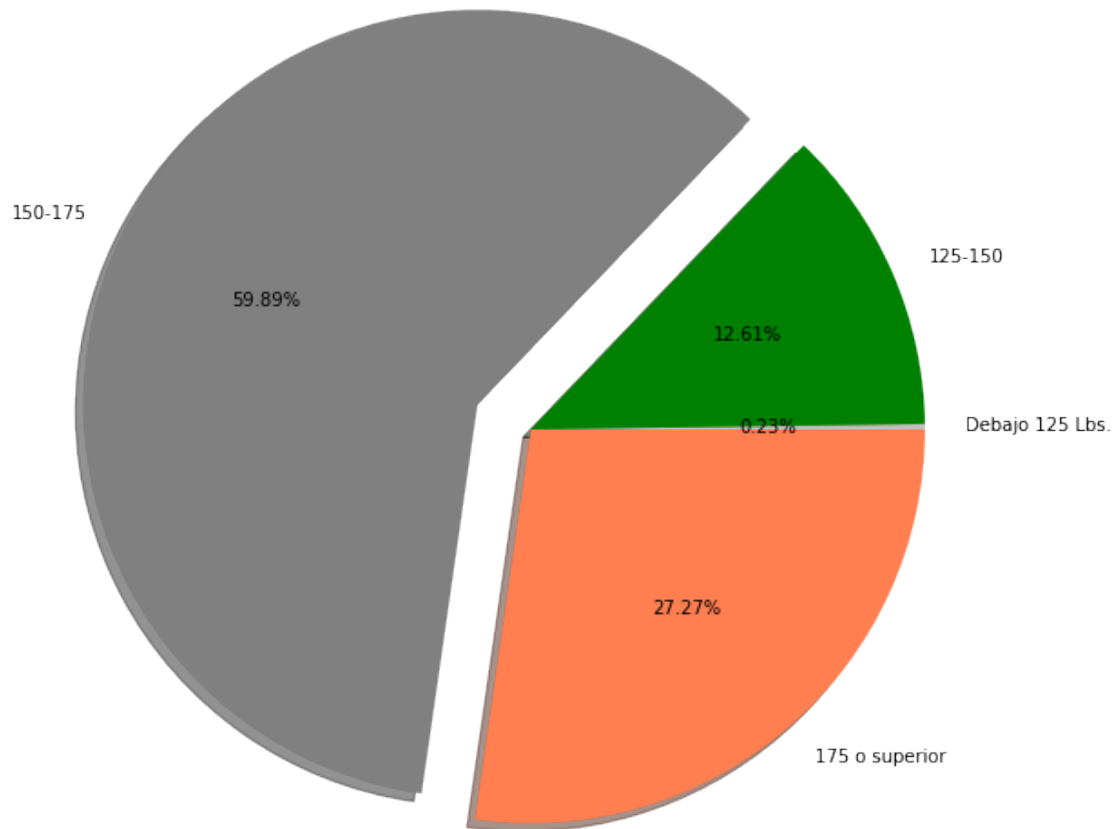
```

rango3 = 0
rango4 = 0
n = 0
for valor in fifa['Weight']:
    if (pd.isna(valor) == True):
        n = n + 1
    elif (float(valor[:3]) <= 125.0):
        rango1 = rango1 + 1
    elif (float(valor[:3]) >= 125.0 and int(valor[:3]) < 150.0):
        rango2 = rango2 + 1
    elif (float(valor[:3]) >= 150.0 and int(valor[:3]) < 175.0):
        rango3 = rango3 + 1
    elif (float(valor[:3]) >= 175.0):
        rango4 = rango4 + 1

pesos = np.array([rango1, rango2, rango3, rango4])
etiquetas = np.array(["Debajo 125 Lbs.", "125-150", "150-175", "175 o
    ↳superior"])
colores = ['silver', 'green', 'grey', 'coral']
explode_vals = [0,0,0.15,0]
plt.pie(pesos, labels=etiquetas, colors = colores, autopct='%1.2f%%',
    ↳shadow=True,
explode = explode_vals)
plt.title('Grafica de los Jugadores segun el rango de edades')
plt.gcf().set_size_inches(50, 10)
plt.show()

```

Grafica de los jugadores segun el rango de edades



```
[12]: #result = result[(result['var']>0.25) | (result['var']<-0.25)]
```

2 Generar un grafico de barras (histograma) de acuerdo a su habilidad (Overall) en base a los siguientes segmentos contando el número de jugadores

- 40
- 50
- 60
- 70
- 80
- 90
- 100

3 Investigar como pasar parametros y generar reportes utilizando Notebook, una de las formas es utilizar papermill

```

[13]: l = []
      for i in fifa['Overall']:
          if i == 40:
              l.append(i)
          elif i == 50:
              l.append(i)
          elif i == 60:
              l.append(i)
          elif i == 70:
              l.append(i)
          elif i == 80:
              l.append(i)
          elif i == 90:
              l.append(i)
          elif i == 100:
              l.append(i)
      intervalos = range(min(l), max(l) + 2)
      plt.hist(x=l, bins=intervalos, color='#F2AB6D', rwidth=0.85)
      plt.gcf().set_size_inches(20, 15)
      plt.title('Histograma de las Habilidades de los Jugadores respecto al Numero de_
      ↪Jugadores')
      plt.xlabel('Overall')
      plt.ylabel('FNumero de Jugadores')
      plt.xticks(intervalos)
      plt.show() #dibujamos el histograma

```

