Practica 02 Simulacion

Carga de los datos de los jugadores de FIFA

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
```

In [2]:
```
df = pd.read_csv('fifa_datos1.csv')
df
```

Out[2]:

	Columna	ID	Name	Age	Photo	Nationality	
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	I
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	I
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	I
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	I
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	
...	...	...	...	...	...	...	
18202	18202	238813	J. Lundstram	19	https://cdn.sofifa.org/players/4/19/238813.png	England	I
18203	18203	243165	N. Christoffersson	19	https://cdn.sofifa.org/players/4/19/243165.png	Sweden	I
18204	18204	241638	B. Worman	16	https://cdn.sofifa.org/players/4/19/241638.png	England	I
18205	18205	246268	D. Walker-Rice	17	https://cdn.sofifa.org/players/4/19/246268.png	England	I
18206	18206	246269	G. Nugent	16	https://cdn.sofifa.org/players/4/19/246269.png	England	I

18207 rows × 89 columns

Con los datos de los jugadores de FIFA, organizarlos de acuerdo al peso (debajo de las 125 lbs, 125 - 150 lbs, 150 - 175 lbs, 175 lbs en adelante) por medio de un gráfico de pie.

In [3]:
```
p1 = 0
p2 = 0
```

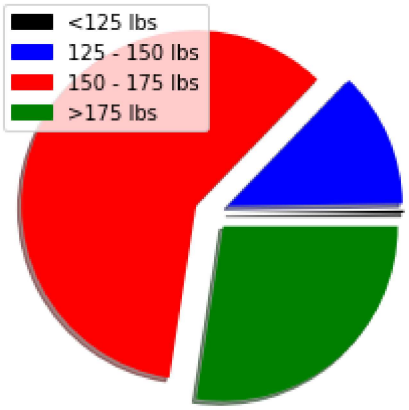
```
p3 = 0
p4 = 0
separador = "lbs"
for x in df['Weight']:
    separado = str(x).split(separador)
    x1 = float(separado[0])
    if x1 < 125:
        p1+=1
    elif x1 < 150:
        p2+=1
    elif x1 < 175:
        p3+=1
    else:
        p4+=1
pesos = [p1, p2, p3, p4]
pesos
```

Out[3]: [41, 2290, 10876, 5000]

In [4]:

```
colors = ['black', 'blue', 'red', 'green']
labels = ['<125 lbs', '125 - 150 lbs', '150 - 175 lbs', '>175 lbs']
myexplode = [0.1, 0.1, 0.1, 0.1]

plt.pie(pesos, colors=colors, explode = myexplode, shadow = True)
lg1 = mpatches.Patch(color='black', label='<125 lbs')
lg2 = mpatches.Patch(color='blue', label='125 - 150 lbs')
lg3 = mpatches.Patch(color='red', label='150 - 175 lbs')
lg4 = mpatches.Patch(color='green', label='>175 lbs')
plt.legend(handles=[lg1, lg2, lg3, lg4])
plt.show()
```



Generar un gráfico de barras de acuerdo al overall de los jugadores en base a los siguientes segmentos:

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

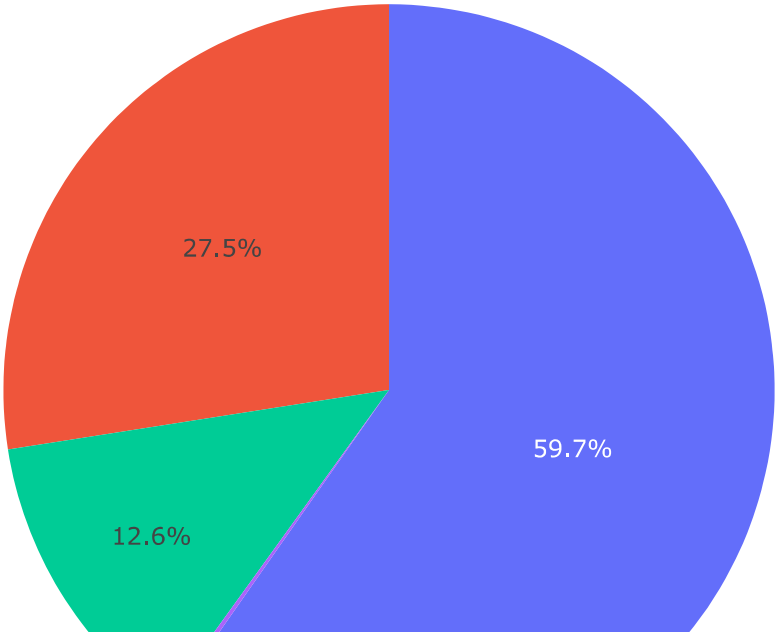
Uso de la libreria Plotly

In [5]:
```
import plotly.graph_objects as go
```

In [6]:
```
import plotly.express as px
import numpy

random_x = [41, 2290, 10876, 5000]
names = ['<125 lbs', '125 - 150 lbs', '150 - 175 lbs', '>175 lbs']

fig = px.pie(values=random_x, names=names)
fig.show()
```



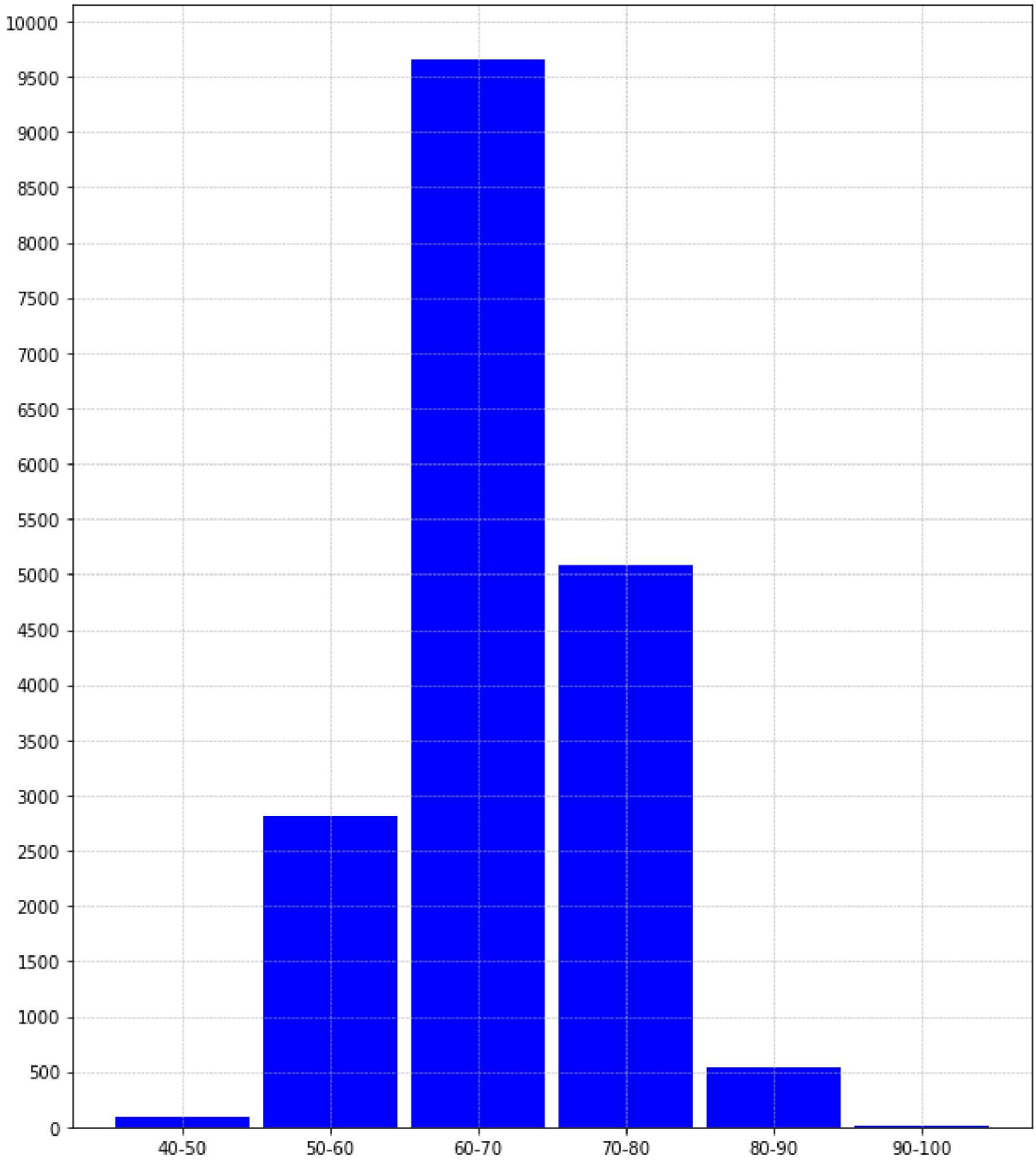
In [7]:
```
x1 = 0; x2 = 0; x3= 0; x4= 0; x5= 0; x6 = 0
for x in df['Overall']:
    if x < 50:
        x1+=1
    elif x < 60:
        x2 +=1
    elif x < 70:
        x3+=1
    elif x < 80:
```

```
        x4+=1
    elif x < 90:
        x5+=1
    else:
        x6+=1
overall = [x1, x2, x3, x4, x5, x6]
overall
```

Out[7]: [89, 2815, 9665, 5083, 541, 14]

In [8]:

```
plt.figure(figsize=(10,12))
plt.bar(['40-50', '50-60', '60-70', '70-80', '80-90', '90-100'], overall, color='blue',
plt.yticks(np.arange(0, 10001, 500))
plt.grid(linestyle='--', linewidth=0.5)
plt.show()
```



Investigacion

Investigar alguna herramienta o software que permita generar resportes con Python utilizando CodeLabs

La herramienta que nos permitira generar reportes con Python se denomina reporlab