



TEMA:

Graficas Fifa y Uso de papermill

Nombre: Esteban David Rosero Pérez

Asignatura:

Simulación

Docente:

Ing. Diego Quisi

Fecha:

Cuenca, 23 de Mayo de 2021

1. Análisis con graficas los datos de FIFA

En el proyecto vamos a graficar los datos de fifa con la librería de matplotlib, en el cual nos facilita a la hora de representar los datos agrupándolos en diferentes formas y de una manera visual poder indicar los factores que tienen relación en los datos.

Los datos que vamos a graficar son sacados de kaggle la cual contiene información sobre los jugadores de importantes equipos de futbol.

1. Importamos Librerias

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

2. Lectura de Archivos

```
df = pd.read_csv('./data.csv')
df.sample(5)
```

5]:

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag	Overall	Potential	Club
14198	14198	223580	R. Miller	29	https://cdn.sofifa.org/players/4/19/223580.png	England	https://cdn.sofifa.org/flags/14.png	61	61	Port Vale
5285	5285	235927	P. Cuadra	23	https://cdn.sofifa.org/players/4/19/235927.png	Argentina	https://cdn.sofifa.org/flags/52.png	70	79	Racing Club
15438	15438	224318	O. Sterling-James	24	https://cdn.sofifa.org/players/4/19/224318.png	St Kitts Nevis	https://cdn.sofifa.org/flags/89.png	59	62	Mansfield Town
9730	9730	51159	Nelson Ferreira	36	https://cdn.sofifa.org/players/4/19/51159.png	Portugal	https://cdn.sofifa.org/flags/38.png	66	66	FC Thun
9959	9959	211720	M. Stiefler	29	https://cdn.sofifa.org/players/4/19/211720.png	Germany	https://cdn.sofifa.org/flags/21.png	65	65	Karlsruher SC

5 rows × 89 columns

Vamos a realizar las graficas para representar el promedio de peso, como también el promedio de las habilidades que posee cada jugador, para ello debemos agrupar en un dataset el nombre con su peso respectivo y en otro dataset el nombre con su promedio de habilidad.

3. Graficas

Primera Grafica

Con los datos del FIFA, organizar a los jugadores de acuerdo al peso en las siguientes escalas

- Debajo 125Lbs
- 125Lbs - 150Lbs
- 150Lbs - 175Lbs
- Superior a 175Lbs

Primero Agrupamos a los jugadores con su respectivo peso

```
] : dfnew = pd.DataFrame(list(zip(df["Name"],df["Weight"].str[0:3])), columns = ['Name','Weight'])
dfnew.head()
```

it[6]:

	Name	Weight
0	L. Messi	159
1	Cristiano Ronaldo	183
2	Neymar Jr	150
3	De Gea	168
4	K. De Bruyne	154

Dividimos a los datos en las categorías indicadas en el Enunciado

```
: ▶ bajo=results[results<=125].size
    medio_bajo=results[np.logical_and(results > 125,results <= 150)].size
    medio_alto=results[np.logical_and(results > 150,results < 175)].size
    alto=results[results>=175].size
```

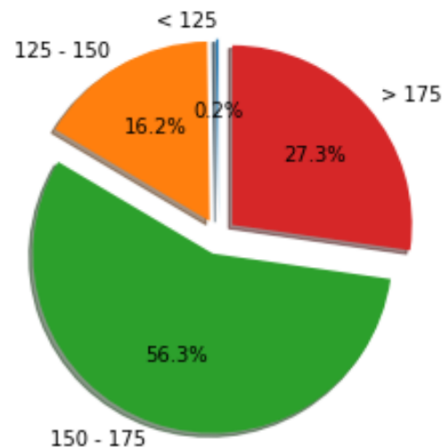
Resultado de la primera grafica

Grafico PIE

```
▶ labels = '< 125', '125 - 150', '150 - 175', '> 175'
  sizes = [bajo, medio_bajo, medio_alto, alto]
  explode = (0.1, 0.1, 0.1, 0.1)

  fig1, ax1 = plt.subplots()
  ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
          shadow=True, startangle=90)
  ax1.axis('equal')

  plt.show()
```



Aquí podemos obtener como resultado que la mayoría de los jugadores tiene un peso promedio de 150 a 175 Lbs seguido de un superior a ese peso, por lo que tenemos que tomar en cuenta que son atletas de máximo nivel que deben cumplir estándares a nivel corporal para poder jugar en el nivel mas alto competitivamente hablando.

Segunda Grafica

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

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

Agrupamos el nombre del jugador con su habilidad

```
dfoverall = pd.DataFrame(list(zip(df["Name"],df["Overall"])), columns = ['Name','Overall'])
dfoverall.head()
```

10]:

	Name	Overall
0	L. Messi	94
1	Cristiano Ronaldo	94
2	Neymar Jr	92
3	De Gea	91
4	K. De Bruyne	91

Después según el enunciado juntamos los promedios de los jugadores en los rangos de 40, 50, 60, 70, 80, 90, 100. Respectivamente por lo que tenemos el siguiente código

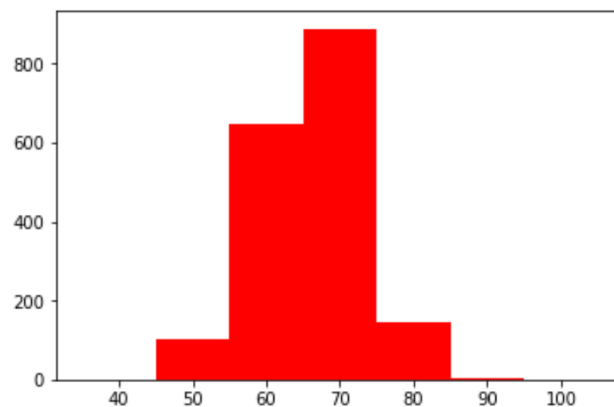
```
cua=results[ov==40].size
cin=results[ov==50].size
ses=results[ov==60].size
sete=results[ov==70].size
oc=results[ov==80].size
nov=results[ov==90].size
cie=results[ov==100].size

total=[cua,cin,ses,sete,oc,nov,cie]
```

Obtenemos los siguientes resultados de las graficas.

Obtenemos las Siguietes Graficas

```
alphab = ['40', '50', '60', '70', '80', '90', '100']
frecuencias = [cu,cinc,ses,sete,oc,nov,cie]
plt.bar(alphab, frecuencias, 1.0, color='r')
plt.show()
```



Como podemos ver el resultado nos indica que un bajo porcentaje de jugadores que tienen una habilidad de 90, el mayor porcentaje de jugadores tienen un promedio de 70, en el cual se puede observar que el promedio de los jugadores esta entre 60 a 70 que son la mayoría de jugadores estándar.

2. Practica de papermill para parametrizar los cuadernos de Jupyter

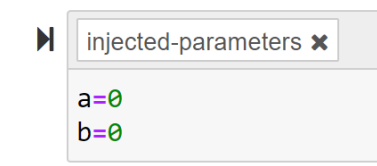
En el mundo de la ciencia de datos para hacer pruebas, ejercicios y documentar proyectos existe jupyter, en el cual es fácil de manipular y nos permite realizar un sin número de prácticas, pero no tiene mucho dinamismo al momento de realizar diferentes pruebas con variables de entrada distintas, para ello surge la idea de papermill que nos permite enviar parámetros desde fuera al momento de ejecutar un cuaderno de Jupyter.

Instalación

Con el siguiente comando se procede a instalar papermill

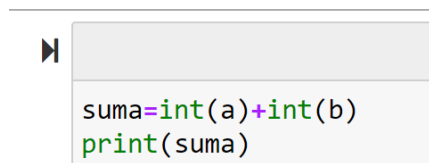
```
Pip install papermill
```

Para poder usarlo debemos tomar en cuenta la creación de un cuaderno en jupyter, en este cuaderno nosotros vamos a incorporar la variables que van a pasar como entrada, en nuestro ejemplo vamos a ver que colocamos las variables a y b, pero el punto de conexión es declarar la celda como injected-parameter en esta declaramos que va a estar a la escucha del envio de variables.



```
injected-parameters x
a=0
b=0
```

Declarado las variables procedemos a realizar el código que teníamos pensado realizar en nuestro caso vamos a realizar una suma como prueba.



```
suma=int(a)+int(b)
print(suma)
```

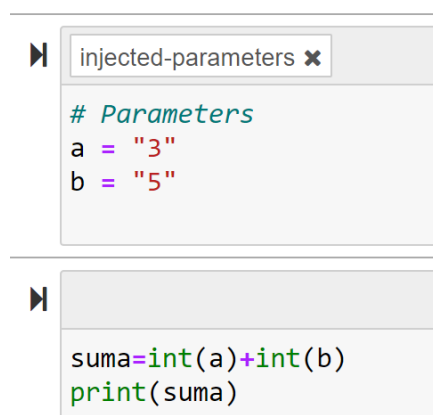
Ahora vamos a crear el llamado de papermill que es un archivo de tipo Python

```
main.py > ...
1  import papermill as pm
2  import argparse
3
4  def main(numeros):
5      pm.execute_notebook('Untitled.ipynb',
6                          'output.ipynb',
7                          parameters=dict(a=numeros[0],b=numeros[1]))
8
9
10 if __name__ == '__main__':
11     args = argparse.ArgumentParser()
12
13     args.add_argument(
14         'int',
15         help='pasar A y B',
16         type=str
17     )
18
19     arguments = args.parse_args()
20
21     main([i for i in arguments.int.split(',')])
```

Con este archivo nosotros lo que hacemos es declarar una función en el cual va a ingresar como parámetros el libro que creamos anteriormente y vamos a declarar la salida, después escribimos que va a ingresar 2 números como argumentos desde la línea de comando, ahora corremos este archivo con el siguiente código.

```
(base) C:\Users\EstebanRM\Desktop\Esteban\universidad\10 semestre\simulacion\tarea grafica\ejemplo>python main.py 3,5
Executing: 100%| 3/3 [00:04<00:00, 1.42s/cell]
```

Y nos entrega un cuaderno de nombre “output.ipynb”



```
# Parameters
a = "3"
b = "5"

suma=int(a)+int(b)
print(suma)
```

8

Como resultado podemos observar que corre todas las celdas que tenía el cuaderno de entrada llamado Untitled.ipynb

Conclusiones:

Matplotlib es una de las librerías de visualización más utilizadas de Python, por lo que nos ayuda a representar datos contenidos en listas, con ello tenemos varias opciones para graficar los datos ya sea un histograma, un gráfico tipo Pie ,etc.

Papermill nos ayuda a parametrizar cuadernos de jupyter haciéndolo más dinámicos, de esta forma podemos ver varias respuestas con diferentes variables corriendo el mismo cuaderno en un tiempo más corto.