

▼ 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.



```
# pip install google-colab
```

```
#IMPORTAR LA LIBRERIAS A USAR
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import math
# SE USA LIBRERIAS DEL SISTEMA
import os
# IMPORTANTE LIBRERIA PARA MONTAR UNIDAD DE GOOGLE
from google.colab import drive
drive.mount('/content/drive/')
#SE CREA PUNTOS DE UNA MATRIZ
x = [0,1,2,3,4]
y = [0,2,4,6,8]
```

Drive already mounted at /content/drive/; to attempt to forcibly remount, call drive

▼ SE CONSTRUYE UN ESTADO DE GRAFICA

Para la contriccion de este primer grafico utilizaremos dos arreglos

```
# SE APLICA UN TAMAÑO Y NUMERO DE PIXELES POR CADA PULGADA.
plt.figure(figsize=(8,5), dpi=250)
# GRAFICOS POR PUNTOS
plt.plot(x,y,'bo')
plt.title('Grafica de puntos', fontdict={'fontname':'Comic Sans MS', 'fontsize': 20})
plt.show()
#Graficamos linea
plt.plot(x,y, color='red', linewidth=2, linestyle='--', label='Linea')
```

```

plt.title('GRAFICAS DE LINEAS RECTAS', fontdict={'fontname': 'Comic Sans MS', 'fontsize':
plt.show()
# 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('Grafica de la ecuacion de la recta', fontdict={'fontname': 'Comic Sans MS', 'fo
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=250)

plt.show()
# Graficar funciones matematicas

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

plt.figure(figsize=(8,5))
plt.title('GRAFICO FUNCION SENO', fontdict={'fontname': 'Comic Sans MS', 'fontsize': 40})
plt.xlabel('Eje X')
plt.ylabel('Eje Y')
plt.plot(x3,y3)
plt.show()
#Grafico de Barras
etiquetas = ['A', 'B', 'C']
valores = np.arange(1,4,1)
plt.figure(figsize=(5,3))
plt.title('GRAFICO BARRAS', fontdict={'fontname': 'Comic Sans MS', 'fontsize': 40})

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

plt.show()

#Trabajar con datos en formato .csv

datos = pd.read_csv('/content/drive/MyDrive/Inteligencia-Artificial-II/simulaciondatos/gas

plt.figure(figsize=(5,3))
plt.title('Reseña del Gas')

```

```
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())

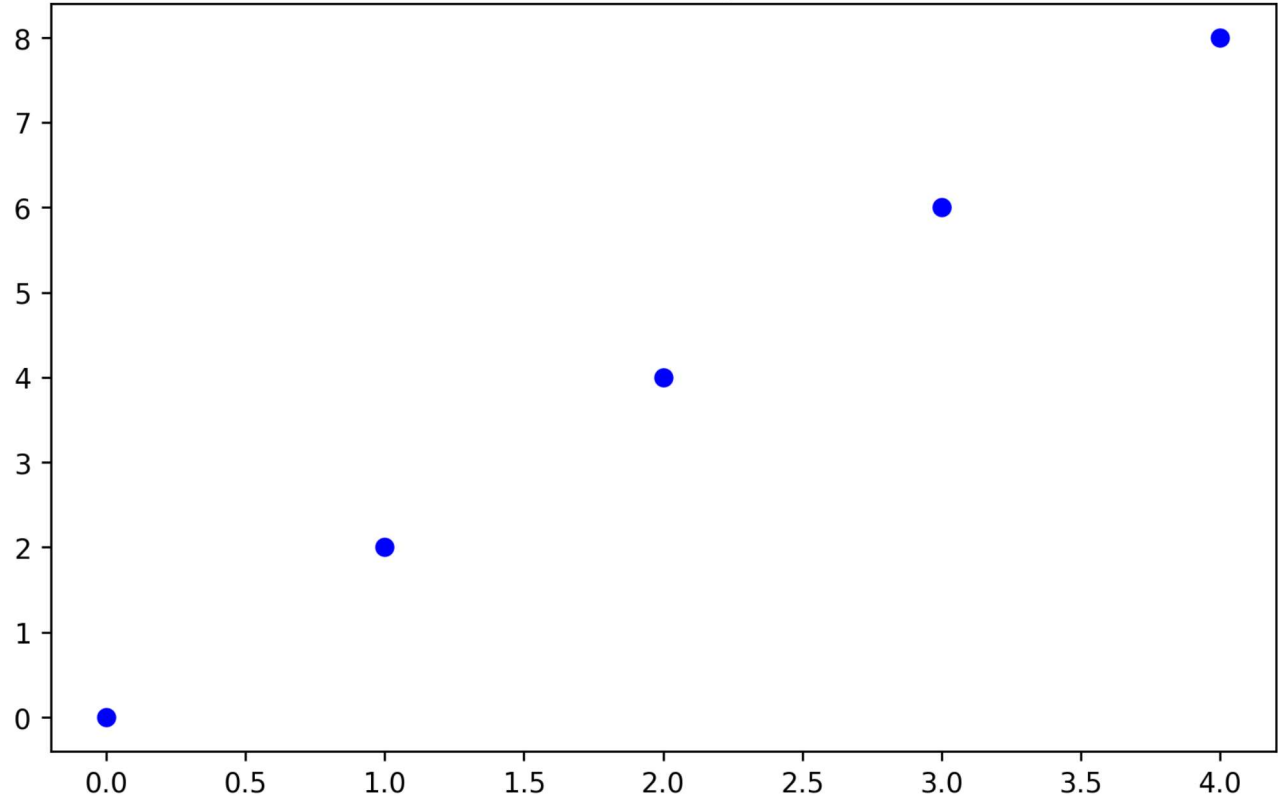
plt.xlabel('Año')
plt.ylabel('Valor Dolares')
plt.legend()
plt.show()
#Trabajar con otro tipo de datos Fifa
#fifa = pd.read_csv('fifa_datos.csv')
fifa = pd.read_csv('/content/drive/MyDrive/Inteligencia-Artificial-II/simulaciondatos/fifa')
#imprimir los primeros 5 datos del archivo
fifa.head(5)
# Generar 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]

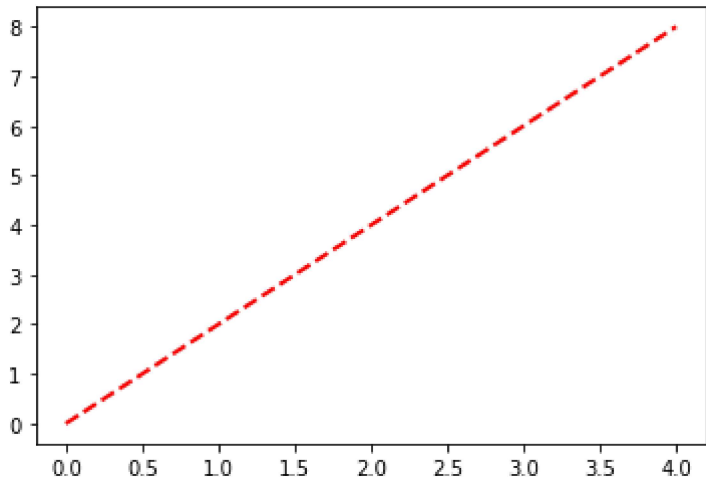
plt.figure(figsize=(8,5))

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

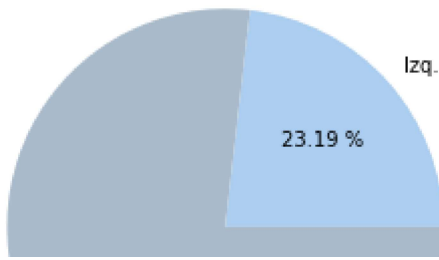
Grafica de puntos



GRAFICAS DE LINEAS RECTAS



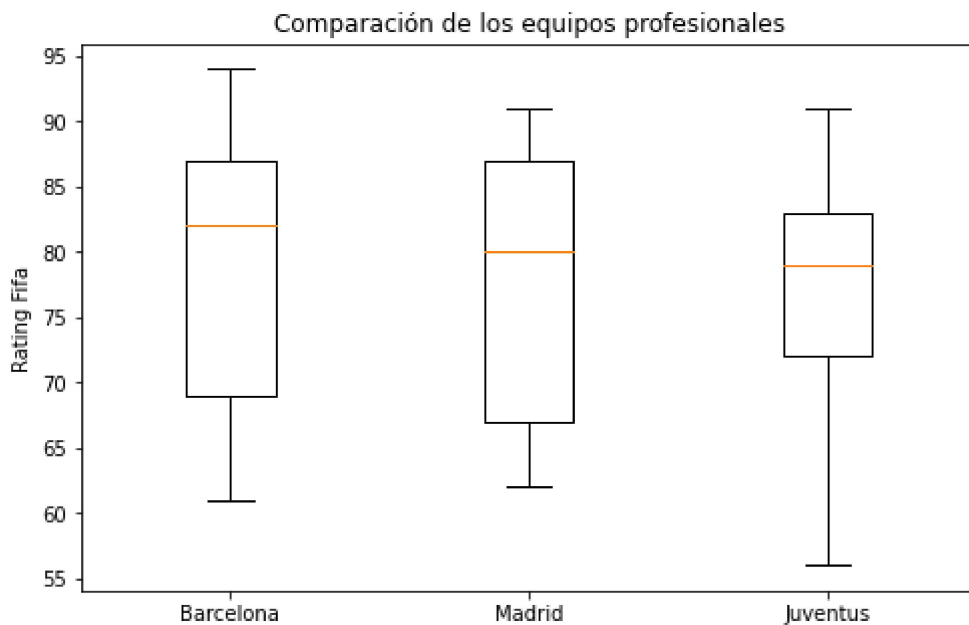
Pie de juego preferido



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()
```



▼ DEBER FIFA

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

nri = 0

```

pri = 0
seg = 0
ter = 0
cua = 0
n = 0
try:
    for i in fifa['Weight']:
        if (pd.isna(i) == True):
            n = n + 1
        elif (float(i[:3]) < 125.0):
            pri = pri + 1
        elif (float(i[:3]) >= 125.0 and int(i[:3]) < 150.0):
            seg = seg + 1
        elif (float(i[:3]) >= 150.0 and int(i[:3]) < 175.0):
            ter = ter + 1
        elif (float(i[:3]) >= 175.0):
            cua = cua + 1
except TypeError:
    pass
b = np.array([pri, seg, ter, cua])
a = np.array(["Debajo 125 Lbs.", "125-150", "150-175", "175 o superior"])
#print(n)
print(pri)
print(seg)
print(ter)
print(cua)
#print(n+pri+seg +ter+cua)
plt.pie(b, labels=a, autopct="%0.1f %%")
plt.gcf().set_size_inches(50, 10)
plt.show()

```

41
2290
10876
4952



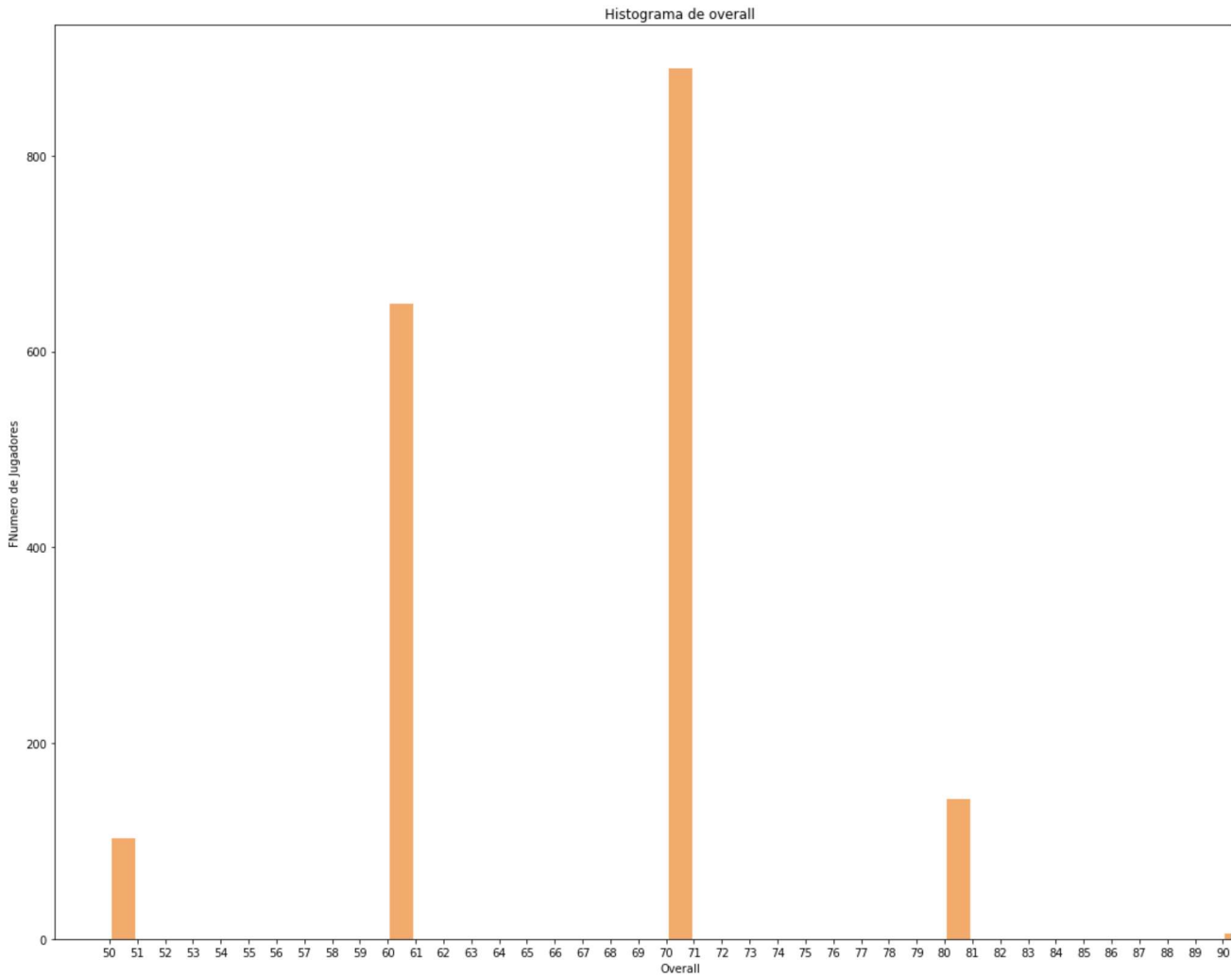
2 Generar un grafico de barras (histograma) de acuerdo a su habilidad (Overall) en base a los siguientes segmentos contando el número de jugadores

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



```
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)
print(l.count(40))
print(l.count(50))
print(l.count(60))
print(l.count(70))
print(l.count(80))
print(l.count(90))
print(l.count(100))
plt.hist(x=l, bins=intervalos, color='#F2AB6D', rwidth=0.85)
plt.gcf().set_size_inches(20, 15)
plt.title('Histograma de overall')
plt.xlabel('Overall')
plt.ylabel('FNumero de Jugadores')
plt.xticks(intervalos)
plt.show() #dibujamos el histograma
```

0
103
649
889
143
5
0



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

