Master Data Science

Aprender a hacer graficas

**Import matplot.pyplot as pt**

**Principales argumentos:**

* **Num :** Numeración de las figuras num = Nones es por defecto y sería el que se haga de forma automática
* **Figsize:** w,h tamaño de la figura en pulgadas
* **Dpi:** Resolución de la imagen en puntos por pulgada
* **Facecolor:** Color del rectángulo
* **Edgecolor:** Color del perímetro
* **Frameon:** si es falso elimina el marco de la figura

**Subplot. Para crear más figuras en misma ventana:**

* **numRows:** numero de filas
* **numCols:** Numero de columnas
* **PlotNum:** Número de gráfica

**Plot Para incluir varias graficas en la misma figura**

* **X =** Abcisas
* **Y =** Ordenandas X e Y tienen que tener el mismo tamaño
* **Linestyle** color y tipo de grafica
* **Linewidth** ancho de línea
* **Marker** marcador

**Tipos de trazos**

**‘-‘** línea solida

‘**-.’** Linea con puntos y rayas

‘- -‘ línea a rayas

‘:’ línea punteada

**Colores**

Los colores son como su inicial en ingles salvo negro que es K

**Ejemplo con leyendas etc:**

Import matplotlib.pyplot as plt

Import numpy as np

N = 8

y = np.zeros(N) # me devuelve un array entre 0 y 8 de ceros.

x1 = np.linspace(0,10, N, endpoint = True) # el cero es el start, el 10 es end, y la N es el numero de intervalos. Endpoint True significa que el último número tiene que ser 10 obligatoriamente.

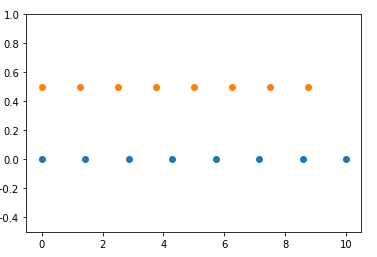
x2 = np.linspace(0, 10, N, endpoint = False)

plt.plot(x1, y, 'o') # la o significa que el tipo de gráficos es de puntos.

plt.plot(x2, y + 0.5, 'o') # si después de la ‘o’ que se podría poner tb market = ‘o’ pongo una s = numero, incrementaría el tamaño de los puntos.

plt.ylim([-0.5,1]) # ylim significa que el eje de la e irá entre -0,5 y 1.

plt.show()



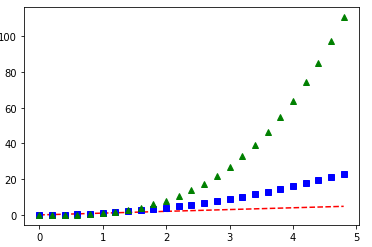
Linspace y arange son parecidos, le decimos el espacio entre los intervalos, pero el primero es mas cuidadoso con que el último número será el definido como end o final .

**Otro ejemplo para los divisores de las lineas**

t = np.arange(0., 5., 0.2) # va de cero a 5 con intervalos de 0,2

plt.plot (t, t, 'r--', t , t\*\*2, 'bs' ,t , t\*\*3, 'g^') # ‘r—’ me devuelve rayas rojas. ‘bs’ me devuelve cuadrados azules ‘g^’ triángulos verdes.

plt.show()



**Ejemplo subplot**

names = ['grupo\_a','grupo\_b','grupo\_c']

values = [1,10,100]

plt.figure(figsize=(9,3)) # Tamaño de los 3 gráficos. Width(anchura), height(altura)

plt.subplot(131) # cada digito tiene un sentido. El 1 son las filas de gráficos, el 3 las columnas de gráficos y el 1 final es el nº de subplot. Se puede ver en la representación del subplot debajo que son 3 graficos en una linea

plt.bar(names,values) # gráfico de barras

plt.subplot(132)

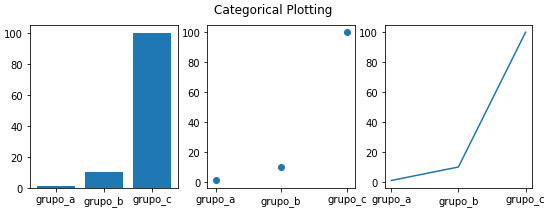
plt.scatter(names,values) # gráfico de puntos. Se suelen utilizar para ver la correlación entre variables.

plt.subplot(133)

plt.plot(names,values) # gráfico de lineas

plt.suptitle('Categorical Plotting') # título de todos los gráficos.

plt.show()



**Ejemplo con múltiples cifras y ejes**

def f(t):

return np.exp(-t) \* np.cos(2\*np.pi\*t)

t1 = np.arange(0.0, 5.0, 0.1)

t2 = np.arange(0.0, 5.0, 0.02)

plt.figure(figsize=(9,7))

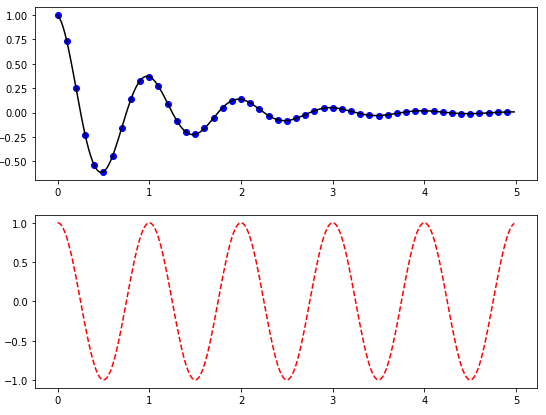
plt.subplot(211)

plt.plot(t1,f(t1),'bo' ,t2, f(t2), 'k')

plt.subplot(212)

plt.plot(t2, np.cos(2\*np.pi\*t2), 'r--')

plt.show()



**Si quiero incluir texto en un grafico utilizo el comando Text. Ejemplo.**

mu , sigma = 100, 15

x = mu + sigma \* np.random.randn(10000)

n, bins, patches = plt.hist(x, 50, density = 1, facecolor = 'g', alpha = 0.75) # bins es el numero de bars que voy a encontrar. Si pongo bins = (10,20,30,40,50) me habría devuelto 5 bars todos los datos entre 10 y 19 en la primera, con los de 20 y 39 en la segunda y así sucesivamente.

plt.xlabel('Smarts')

plt.ylabel('Probability')

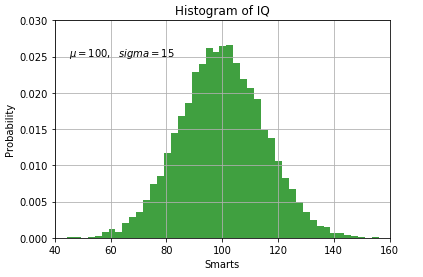
plt.title('Histogram of IQ')

plt.text(45, .025, r'$\mu=100, \ \ sigma = 15$') # Para incluir texto dentro del gráfico

plt.axis([40, 160, 0, 0.03])

plt.grid(True) # para quitar las líneas del fondo tendría que poner False.

plt.show()



**Puedo incluir leyendas dentro del gráfico para saber a qué corresponde cada línea si tengo más de una.**

**Para eso:**

Plt.plot(x,y, label = ‘name’)

Plt.plot(x2,y2, label = ‘name2’)

Plt.legend()

**Ejemplo stack. Es util cuando tengo algo global que se reparte entre varias variables y quiero ver como se reparte entre las variables.**

days = [1,2,3,4,5]

sleeping = [7, 8, 6, 11, 7]

eating = [2, 3, 4, 3, 2]

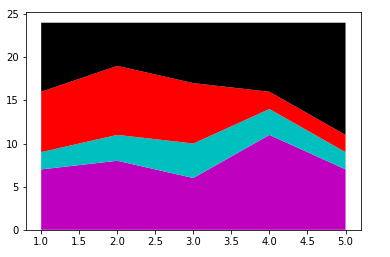
working = [7, 8, 7, 2, 2]

playing = [8, 5, 7, 8, 13]

plt.stackplot(days, sleeping, eating, working, playing, colors = ['m', 'c', 'r', 'k'], labels = ['sleeping','eating', 'working', 'playing'])

NO me devuelve los label

Muy útil para ver en qué dedico mi tiempo cada día.



**Esto también se puede poner con gráfico Pie**

days = [1,2,3,4,5]

sleeping = [7, 8, 6, 11, 7]

eating = [2, 3, 4, 3, 2]

working = [7, 8, 7, 2, 2]

playing = [8, 5, 7, 8, 13]

slices = [7, 2, 2, 13]

activities = ['sleeping', 'eating', 'working', 'playing']

cols = ['m', 'c', 'r', 'b']

plt.pie (slices,

labels = activities,

colors = cols,

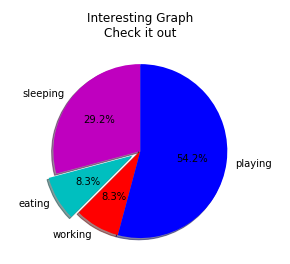
startangle = 90, # es el Angulo desde el que comienza el grafico. La línea empieza justo debajo del titulo.Perpendicular al titulo.

shadow = True, # lo hace un poco más bonito con algo de sombra

explode = (0,0.1,0,0), # saca una de las partes hacia fuera un poco (0.1). El resto son ceros porque no quiero que sobresalgan.

autopct = '%1.1f%%') # incluye el porcentaje dentro de cada apartado.lo calcula aautomaticamente.No tenemos que hacer nosotros el calculo.

plt.title('Interesting Graph\nCheck it out')



**Ejemplo de plot con un fichero CSV**

**Destaco:**

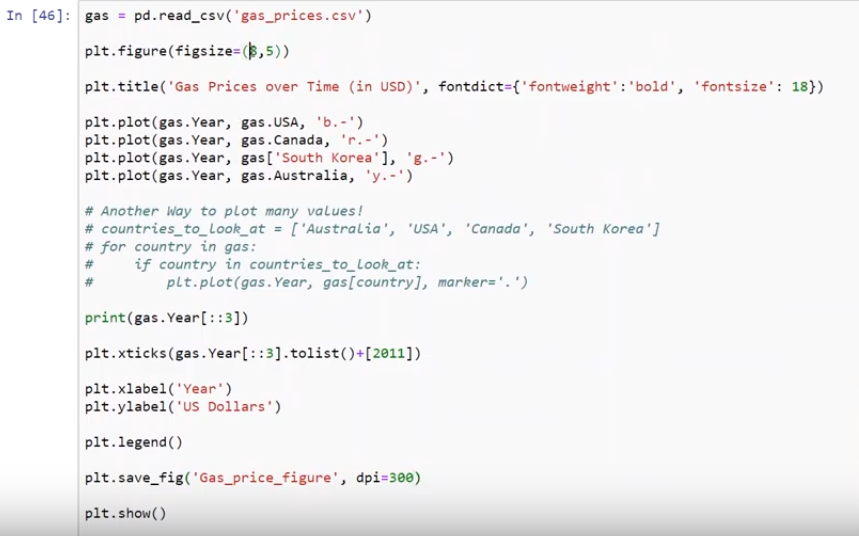
**Fontdict** para cambiar la letra del título aunque podría hacerlo en cualquier texto. En este caso lo hemos puesto en negrita y más grande(fontweight y fontsize)

**For:** Si tenemos muchos campos puedo utilizar un bucle para incluirles en el gráfico y siempre es útil crear una variable por fuera donde fácilmente pueda modificar los campos en un futuro . En este caso 4 países.

**plt.xticks(gas.Tear[::3])** en el grafico salían en el eje de las x todos los años y eso hacían que no se vieran porque eran demasiados. De esta forma sale solo uno de cada 3 años. Muy útil también

**plt.xticks(gas.Tear[::3].tolist()+[2011])** necesito hacer un tolist si quiero añadir un año más al final.

**Plt.savefig(‘Gas\_price\_figure’,dpi = 300)** Es la forma de guardar el gráfico en la misma carpeta que tengo el proyecto. La cifra implicará más resolución cuanto más grande sea. En este caso teníamos figsize (8,5) entonces dpi multiplica por 300 esas cifras de resolución.



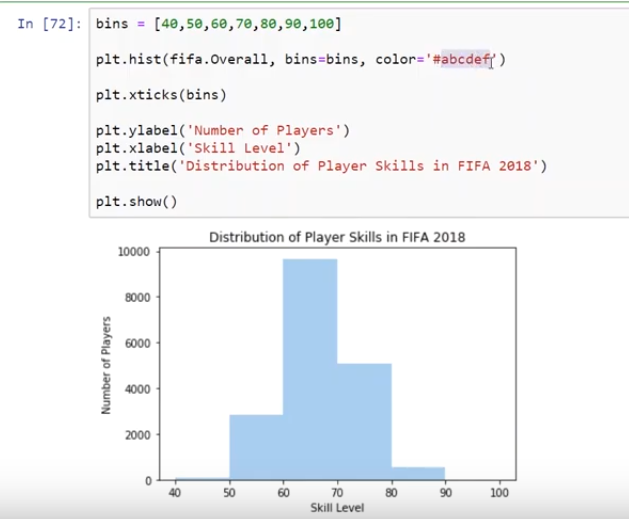
**En el siguiente ejemplo, vemos cómo hacer una selección de un fichero dentro de una columna**

Tomo los jugadores zurdos o diestros de un fichero que se llama fifa. Los estoy contando y por eso tengo que usar count y solo quiero la cifra por eso pongo [0]



**En el siguiente ejemplo veremos de nuevo el uso de bins y como mirando en internet un código puedo poner cualquier color (poner en Google color picker).**

**EN este ejemplo estoy tomando la columna Overall que es donde está la info del score que le da a cada jugador el juego de la fifa.**



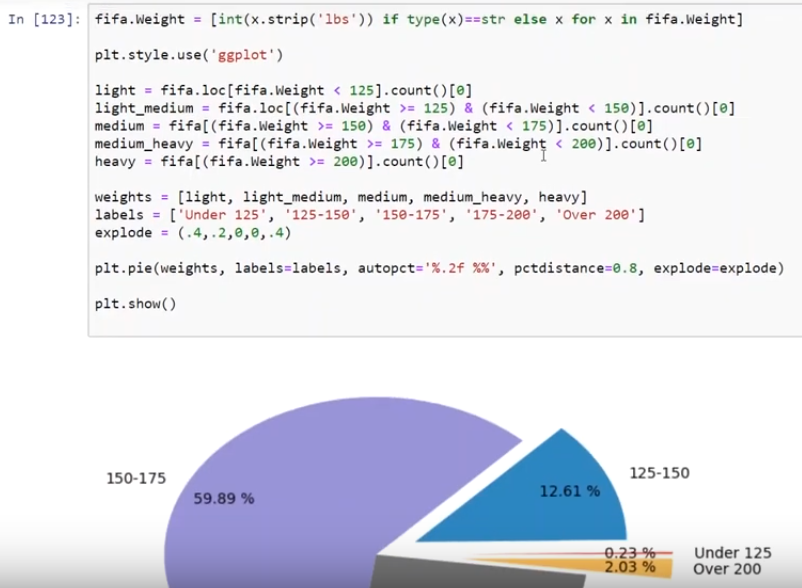
**En el siguiente ejemplo vamos a ver cómo hago secciones de una columna para agruparlas por pesos en este caso.**

**Plt.style.use(‘ggplot):** es para que no tome los colores por defecto.

Las siguientes 5 lineas agrupan los distintos jugadores por peso. **Rangos**

**Explode** como vimos antes separa el grafico si loas cifras salen muy pegadas

**Pctdistance** marca la distancia del centro a los porcentajes. De esa forma se verán mejor si estuvieran muy pegados o incluso sacarlos del gráfico.



**Ejemplo de box plot:**

Selecciono los datos de 3 equipos. En este caso la valoración de sus jugadores en juego fifa.

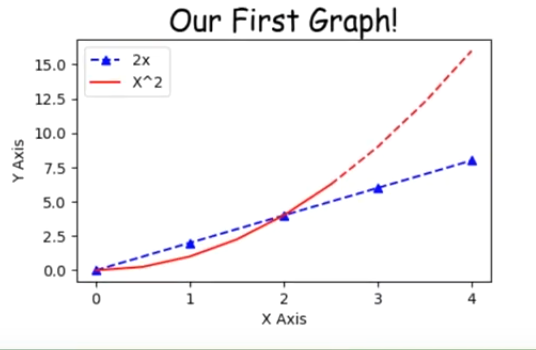
Con for box… cambio el color y la línea. Para el fondo necesito facecolor y en el apartado de boxes patch\_artist = True o te dará error.



**Si quiero hacer un grafico con una estimación futura**

**Se hace a través del plt.plot(x2[:6] el 6 se puede cambiar según los datos que tengas**

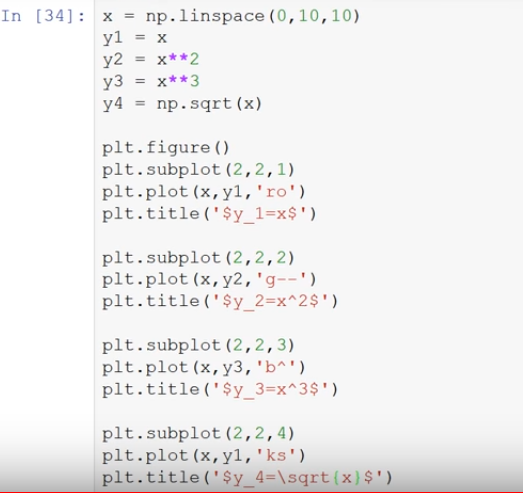


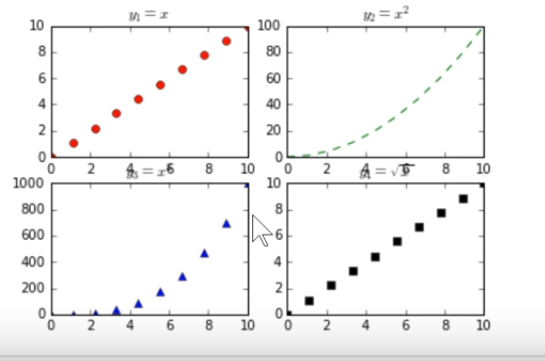


**Ejemplo de subplot**

**Plt.subplt(2,2,1)** los dos primeros números son la posición de gráficos y tienen que ser iguales en todos los subplots. EN este caso 2x2(2 filas y 2 columnas). Si hubiera puesto 4,1 tendría una sola columna con los 4 gráficos- el último 1 es el número de gráfico

En los títulos si pongo el símbolo de **dólar** delante y detrás parecerá más profesional





**Otro ejemplo de subplot**

import matplotlib.pyplot as plt

x = ([1,2,3,4,5])

y1 = ([3,4,5,6,7])

y2 = ([4,2,3,6,7])

plt.figure(figsize=(9,3))

plt.ylim()

plt.subplot(1,2,1)

plt.plot(x,y1,'k--')

plt.title('Primer grafico\n', fontweight = 'bold',fontsize = 10)

plt.xlabel('eje x')

plt.ylabel('eje y')

plt.subplot(1,2,2)

plt.plot(x, y2, 'ro')

plt.title('Segundo grafico\n', fontweight = 'bold', fontsize = 10)

plt.xlabel('eje x')

plt.ylabel('eje y')

plt.show()

