Colores y estilos

In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [ ]:
```
#Declarando x
x = np.linspace(0,5,10)
```

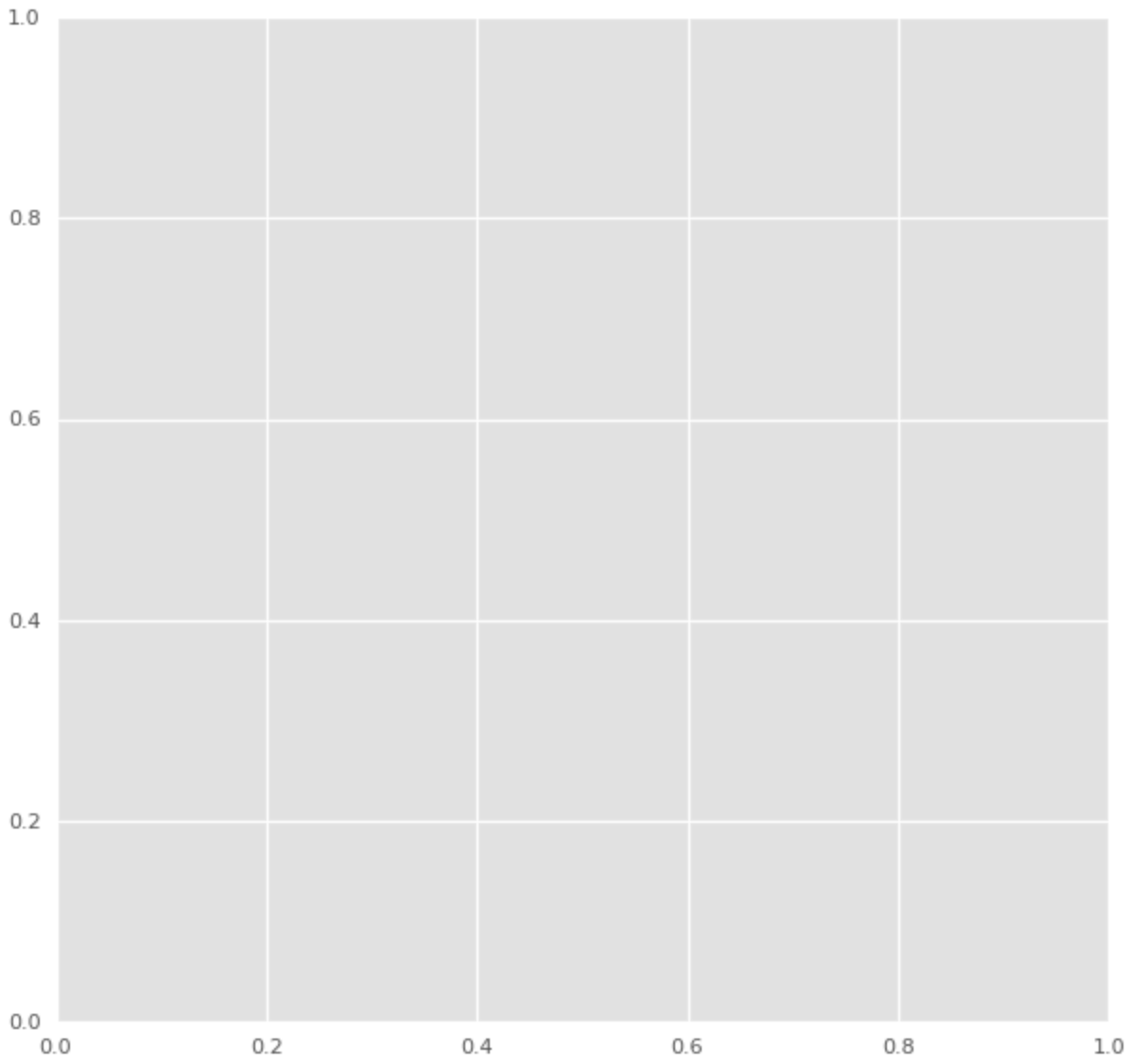
In [ ]:
```
#Desplegando stilos disponibles
for index,estilo in enumerate(plt.style.available):
    print(f'Estilo {[index+1]} = {estilo}')
```

```
Estilo [1] = Solarize_Light2
Estilo [2] = _classic_test_patch
Estilo [3] = _mpl-gallery
Estilo [4] = _mpl-gallery-nogrid
Estilo [5] = bmh
Estilo [6] = classic
Estilo [7] = dark_background
Estilo [8] = fast
Estilo [9] = fivethirtyeight
Estilo [10] = ggplot
Estilo [11] = grayscale
Estilo [12] = seaborn-v0_8
Estilo [13] = seaborn-v0_8-bright
Estilo [14] = seaborn-v0_8-colorblind
Estilo [15] = seaborn-v0_8-dark
Estilo [16] = seaborn-v0_8-dark-palette
Estilo [17] = seaborn-v0_8-darkgrid
Estilo [18] = seaborn-v0_8-deep
Estilo [19] = seaborn-v0_8-muted
Estilo [20] = seaborn-v0_8-notebook
Estilo [21] = seaborn-v0_8-paper
Estilo [22] = seaborn-v0_8-pastel
Estilo [23] = seaborn-v0_8-poster
Estilo [24] = seaborn-v0_8-talk
Estilo [25] = seaborn-v0_8-ticks
Estilo [26] = seaborn-v0_8-white
Estilo [27] = seaborn-v0_8-whitegrid
Estilo [28] = tableau-colorblind10
```

Ver estilos disponibles

```
for i in plt.style.available:
  plt.style.use(i)
  fig, ax = plt.subplots(figsize=(5,5))
  ax.plot(x, y)
```

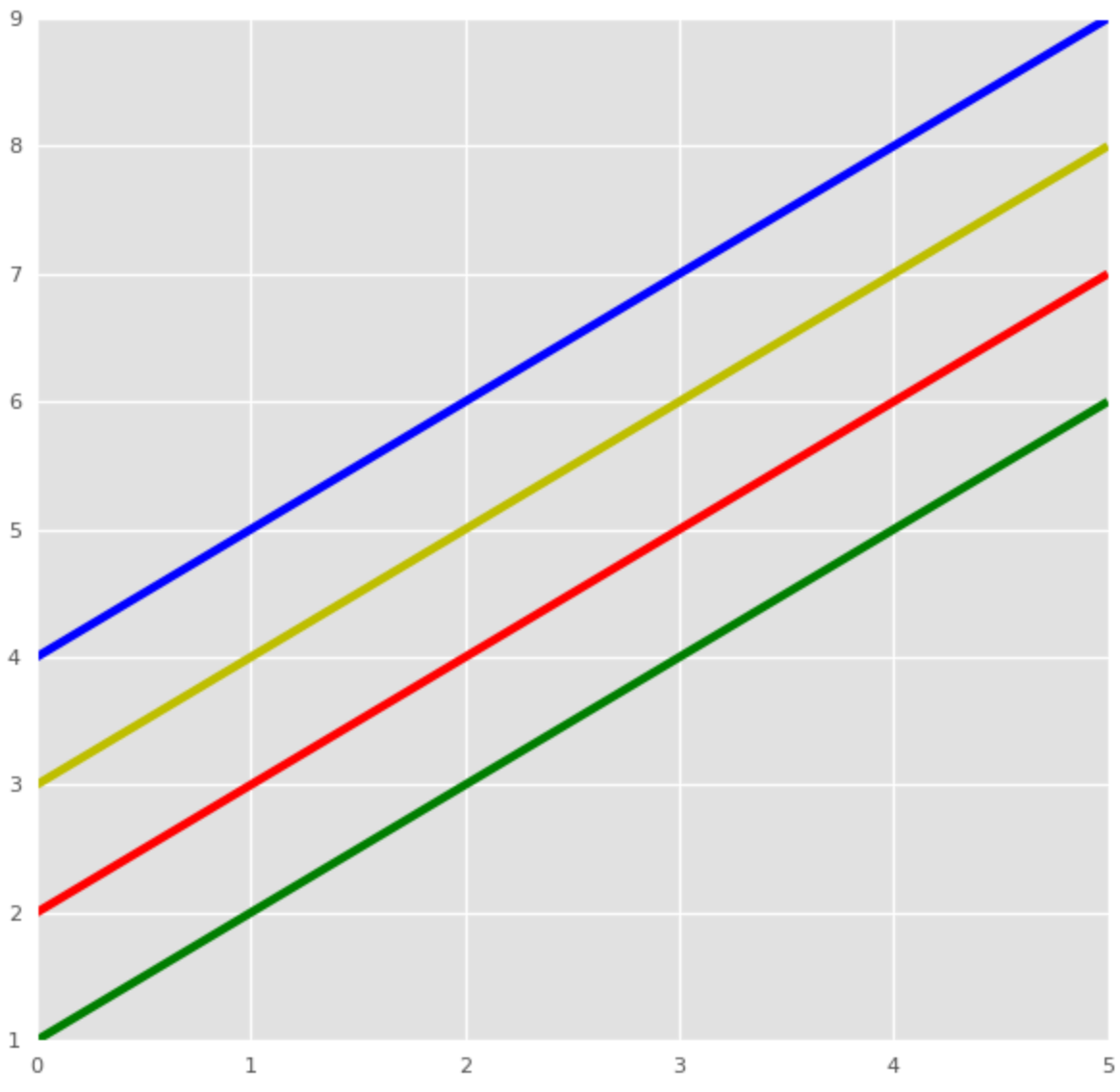
In [ ]:
```
#Usando estilos
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))
```



Definiendo y usando estilos

In [ ]:
```
#Definiendo estilos
#usaré tableau-colorblind10
plt.style.use('tableau-colorblind10')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,'g')
axes.plot(x,x+2,'r')
axes.plot(x,x+3,'y')
axes.plot(x,x+4,'b')
plt.show()
```



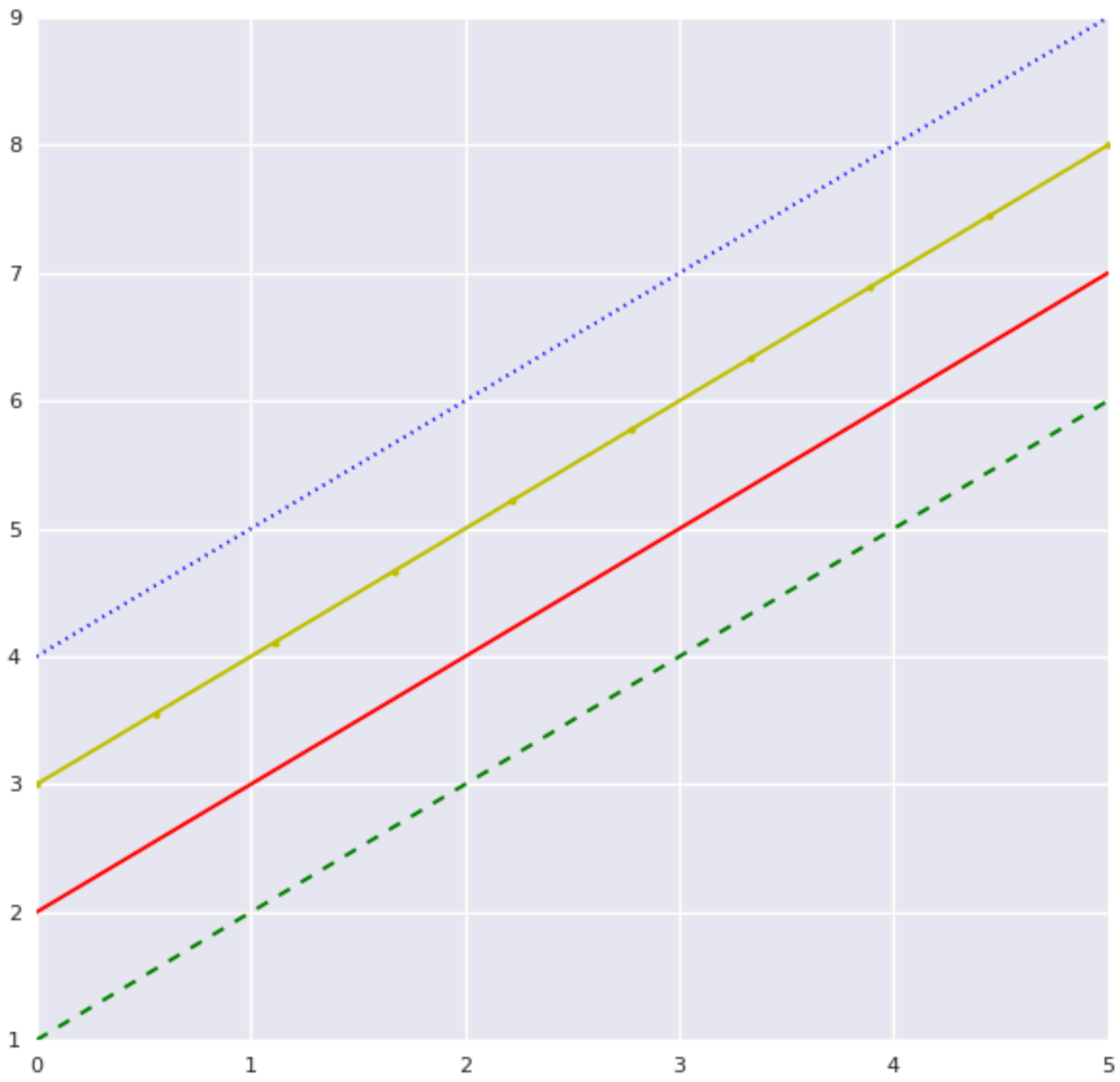
Explicación

Para poder aplicar ciertos estilos a los gráficos, tenemos que usar algo similar como en Matlab

`r--` : Color y estilo de línea

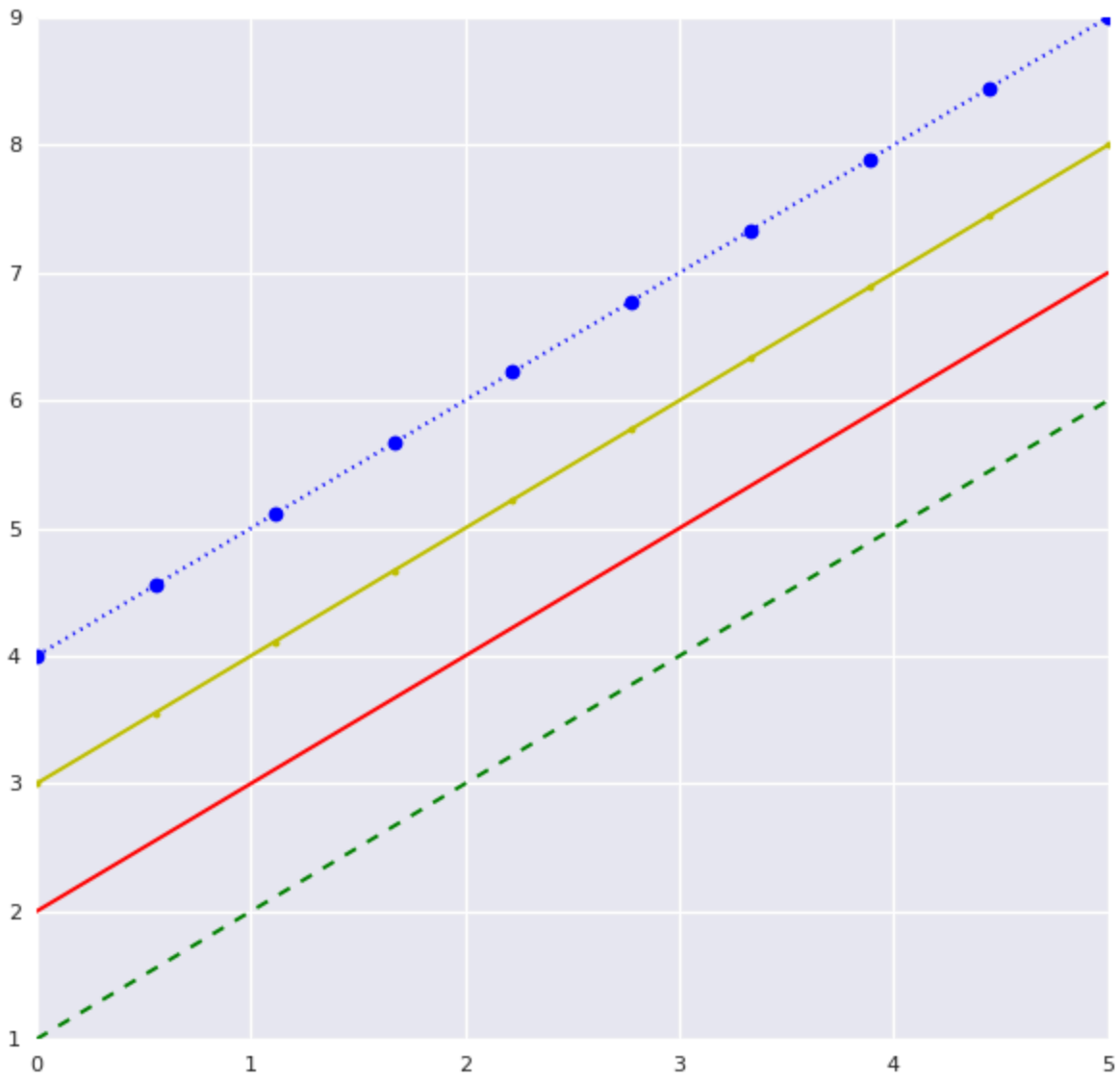
In [ ]:
```
#seaborn-v0_8
plt.style.use('seaborn-v0_8')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,'g--')
axes.plot(x,x+2,'r-')
axes.plot(x,x+3,'y.-')
axes.plot(x,x+4,'b:')
plt.show()
```



In [ ]:
```
#seaborn-v0_8
plt.style.use('seaborn-v0_8')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,'g--')
axes.plot(x,x+2,'r-')
axes.plot(x,x+3,'y.-')
axes.plot(x,x+4,'bo:')
plt.show()
```



Pyplot

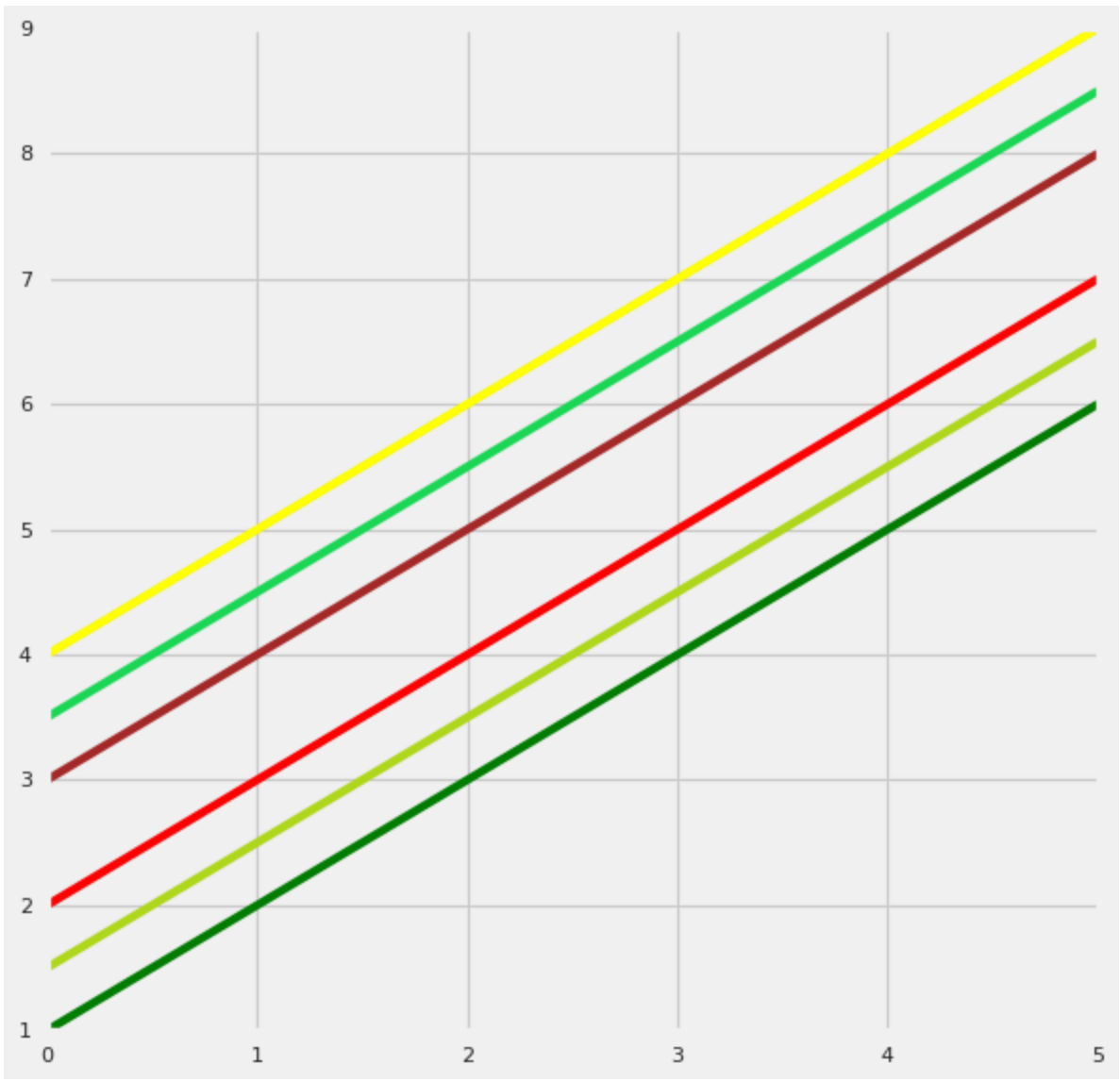
También se puede cambiar y personalizar los colores y las diferentes lineas, pero sin ocupar una estructura similar a la de Matlab.

In [ ]:
```
#fivethirtyeight
plt.style.use('fivethirtyeight')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,color='green')
axes.plot(x,x+2,color='red')
axes.plot(x,x+3,color='brown')
axes.plot(x,x+4,color='yellow')

#Tambien puedo usar colores RGB con Hexadecimal
axes.plot(x,x+1.5,color='#B1D71A')
axes.plot(x,x+3.5,color='#1AD757')

plt.show()
```



Ademas podemos:

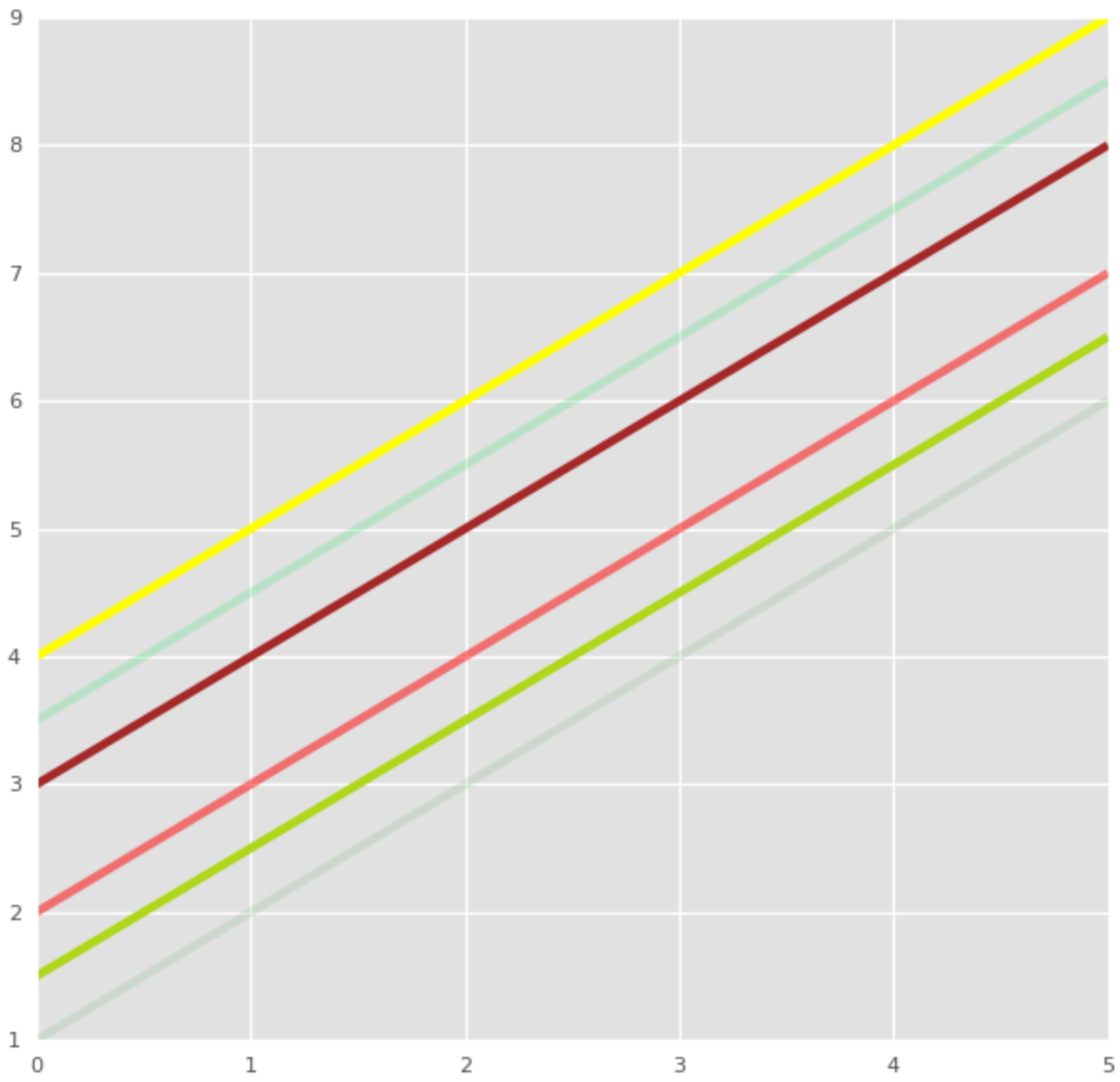
Modificar nivel de transparencia con el canal alpha en las lineas
Veamoslo:

In [ ]:
```
#ggplot
plt.style.use('ggplot')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,color='green',alpha=0.1)
axes.plot(x,x+2,color='red',alpha=.5)
axes.plot(x,x+3,color='brown')
axes.plot(x,x+4,color='yellow')

#Tambien puedo usar colores RGB con Hexadecimal
axes.plot(x,x+1.5,color='#B1D71A')
axes.plot(x,x+3.5,color='#1AD757',alpha=0.2)

plt.show()
```



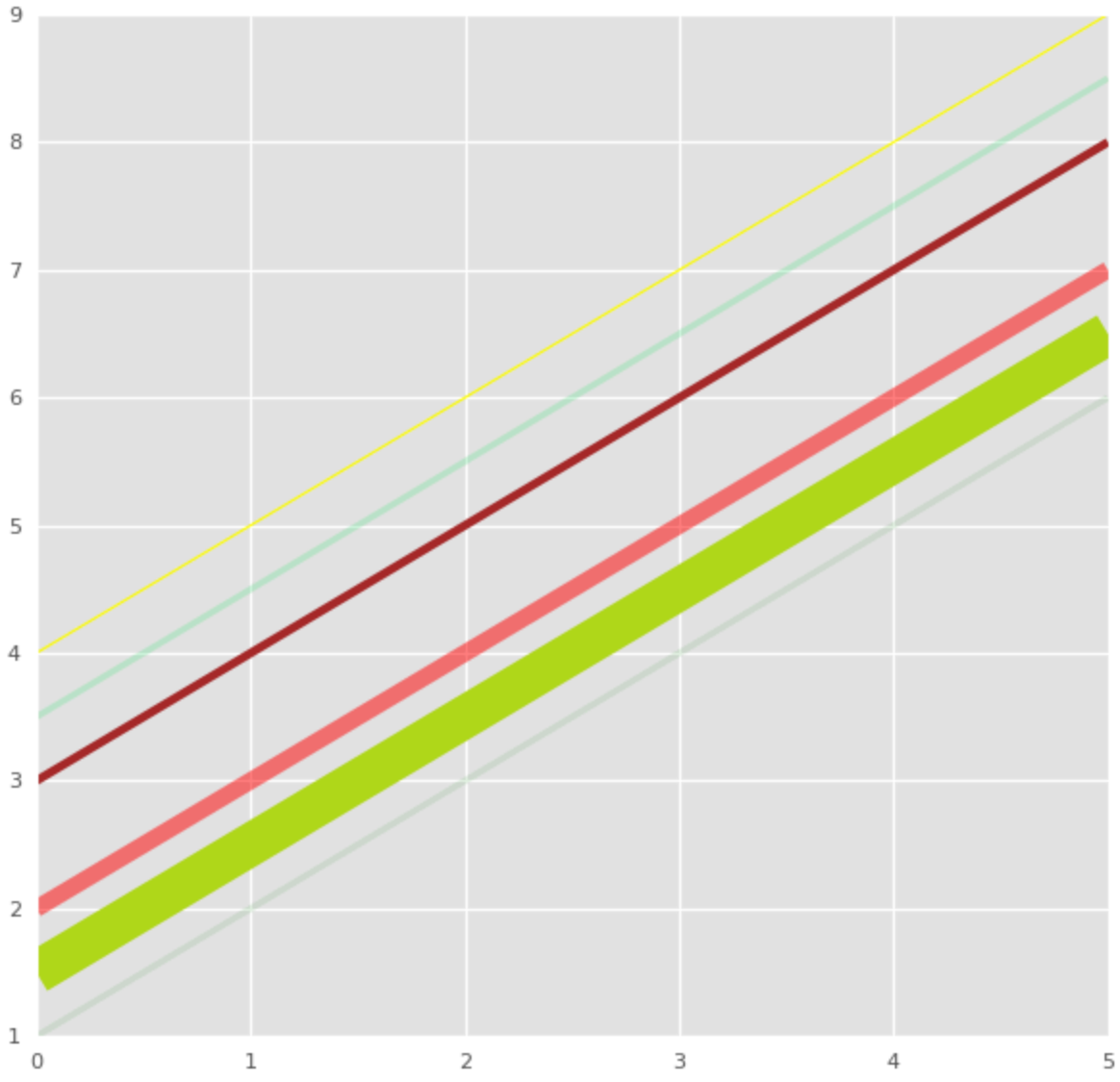
Modificando grosor de linea

In [ ]:
```
#ggplot
plt.style.use('ggplot')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,color='green',alpha=0.1,linewidth=3)
axes.plot(x,x+2,color='red',alpha=.5,linewidth=8)
axes.plot(x,x+3,color='brown')
axes.plot(x,x+4,color='yellow',linewidth=1)

#Tambien puedo usar colores RGB con Hexadecimal
axes.plot(x,x+1.5,color='#B1D71A',linewidth=20)
axes.plot(x,x+3.5,color='#1AD757',alpha=0.2,linewidth=3)

plt.show()
```



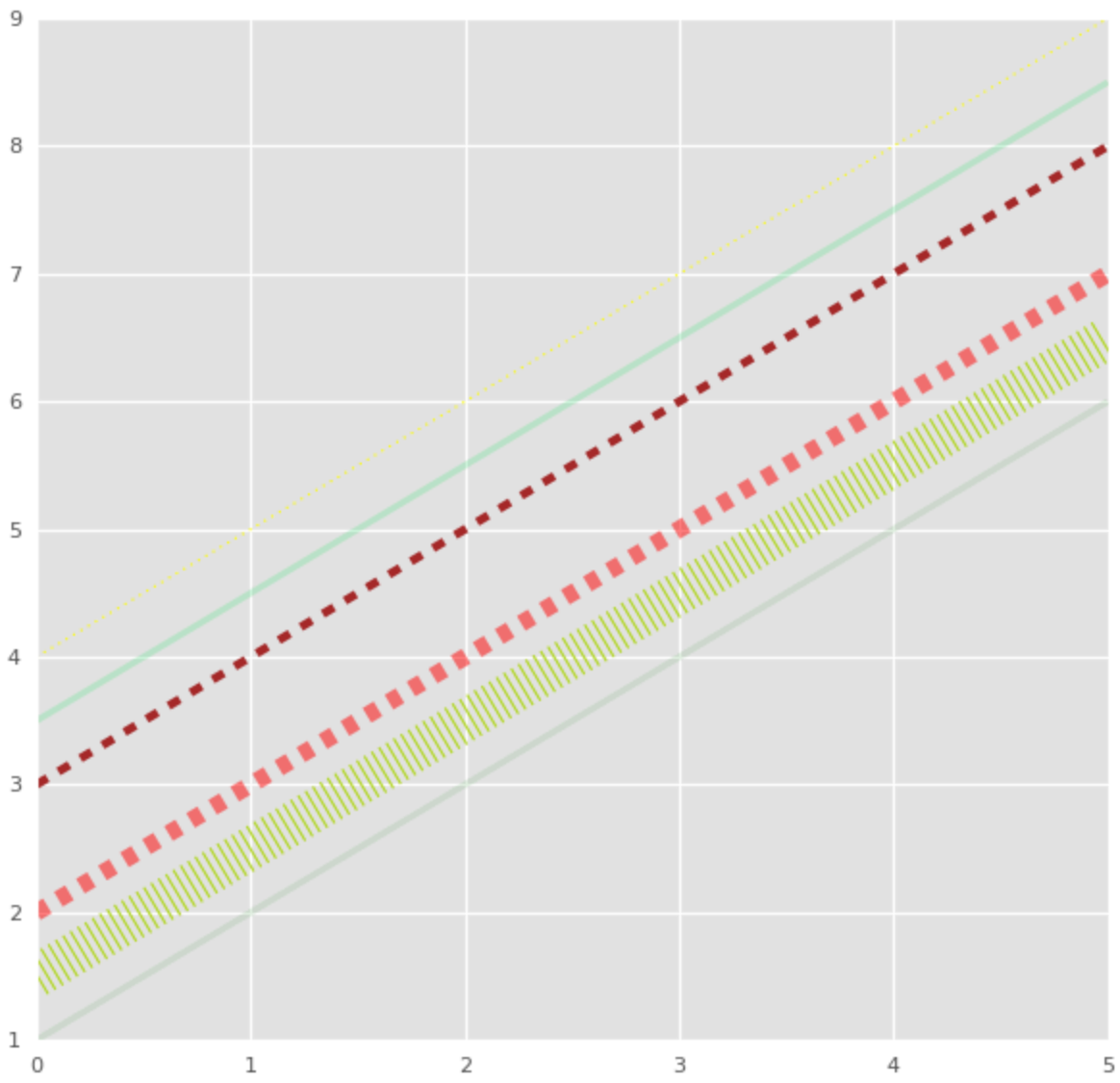
Modificando estilo de linea*

In [ ]:
```
#ggplot
plt.style.use('ggplot')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,color='green',alpha=0.1,linewidth=3,linestyle='-')
axes.plot(x,x+2,color='red',alpha=.5,linewidth=8,linestyle='--')
axes.plot(x,x+3,color='brown',linestyle='dashed')
axes.plot(x,x+4,color='yellow',linewidth=1,linestyle=':')

#Tambien puedo usar colores RGB con Hexadecimal
axes.plot(x,x+1.5,color='#B1D71A',linewidth=20,linestyle='dotted')
axes.plot(x,x+3.5,color='#1AD757',alpha=0.2,linewidth=3)

plt.show()
```



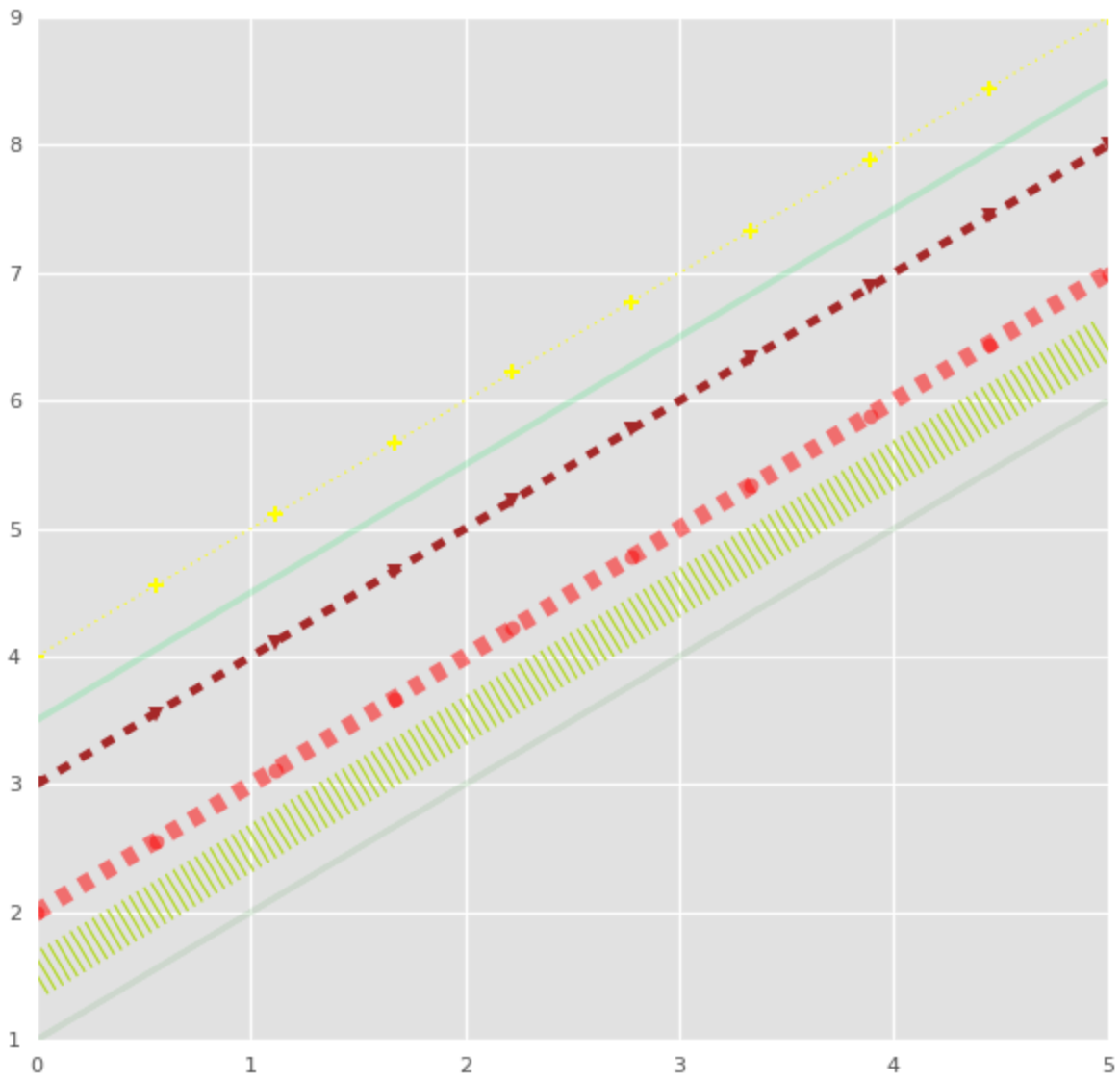
Markers

In [ ]:
```
#ggplot
plt.style.use('ggplot')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,color='green',alpha=0.1,linewidth=3,linestyle='-',marker = 'x')
axes.plot(x,x+2,color='red',alpha=.5,linewidth=8,linestyle='--',marker = 'o')
axes.plot(x,x+3,color='brown',linestyle='dashed',marker = 'v')
axes.plot(x,x+4,color='yellow',linewidth=1,linestyle=':',marker = 'P')

#Tambien puedo usar colores RGB con Hexadecimal
axes.plot(x,x+1.5,color='#B1D71A',linewidth=20,linestyle='dotted')
axes.plot(x,x+3.5,color='#1AD757',alpha=0.2,linewidth=3)

plt.show()
```



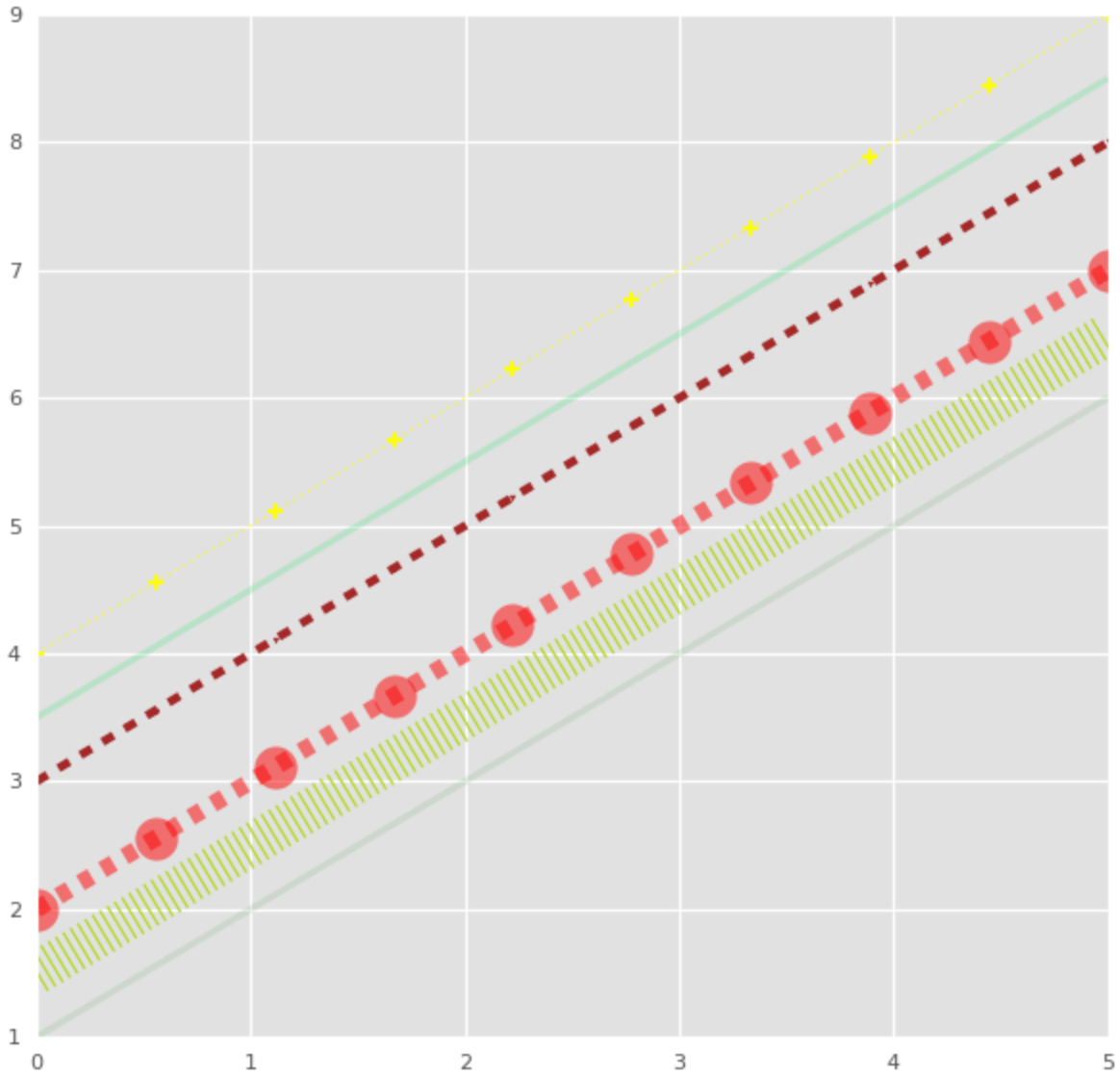
Tamaño del marker

In [ ]:
```
#ggplot
plt.style.use('ggplot')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,color='green',alpha=0.1,linewidth=3,linestyle='-',marker = 'x',markersize=10)
axes.plot(x,x+2,color='red',alpha=.5,linewidth=8,linestyle='--',marker = 'o',markersize=20)
axes.plot(x,x+3,color='brown',linestyle='dashed',marker = 'v',markersize=3)
axes.plot(x,x+4,color='yellow',linewidth=1,linestyle=':',marker = 'P')

#Tambien puedo usar colores RGB con Hexadecimal
axes.plot(x,x+1.5,color='#B1D71A',linewidth=20,linestyle='dotted')
axes.plot(x,x+3.5,color='#1AD757',alpha=0.2,linewidth=3,markersize=10)

plt.show()
```



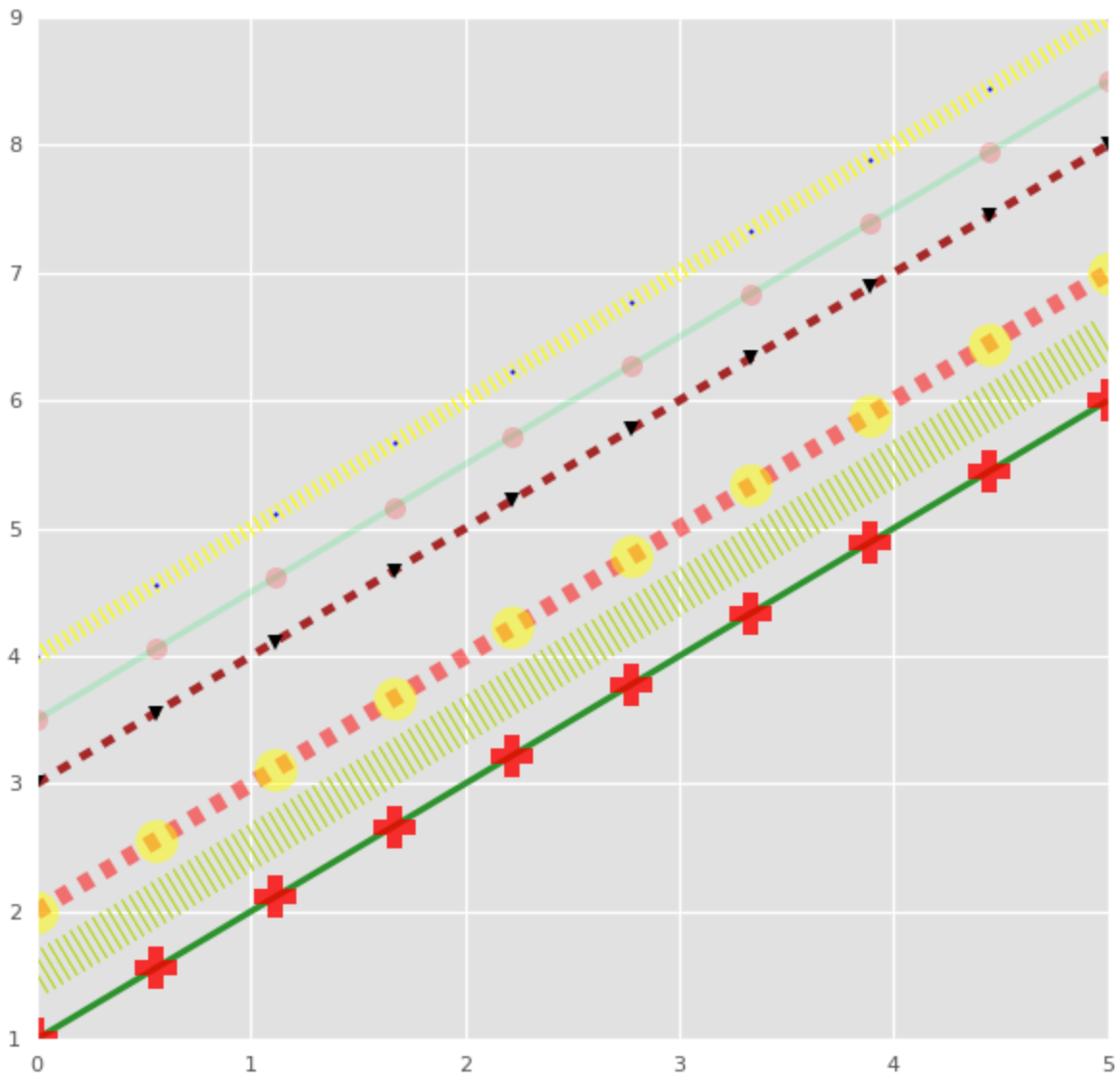
Color de marker

In [ ]:
```
#ggplot
plt.style.use('ggplot')
#Creando lienzo
fig,axes = plt.subplots(figsize=(8,8))

#Graficando
axes.plot(x,x+1,color='green',alpha=0.8,linewidth=3,linestyle='-',marker = 'P',markersize=20,markerfacecolor='red')
axes.plot(x,x+2,color='red',alpha=.5,linewidth=8,linestyle='--',marker = 'o',markersize=20,markerfacecolor='yellow')
axes.plot(x,x+3,color='brown',linestyle='dashed',marker = 'v',markersize=8,markerfacecolor='black')
axes.plot(x,x+4,color='yellow',linewidth=8,linestyle=':',marker = 'o',markersize=2, markerfacecolor='blue')

#Tambien puedo usar colores RGB con Hexadecimal
axes.plot(x,x+1.5,color='#B1D71A',linewidth=20,linestyle='dotted')
axes.plot(x,x+3.5,color='#1AD757',alpha=0.2,linewidth=3,marker = 'o',markersize=10,markerfacecolor='red')
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x7fe148fab350>]
```



NOTA IMPORTANTE:

La calidad de la gráfica y como se representa está directamente relacionada con la cantidad de muestras tomadas.

Si se desea que los marcadores tengan un espaciado correcto. Definamos nuestras variables con la menor cantidad de datos posibles o muestras.

Así:

```
#Declarando x x = np.linspace(0,5,10)
```

> Nos facilitará ver los markers entre las gráficas.

```
#Declarando x x = np.linspace(0,5,100)
```

> Los markers estarán muy proximos, aunque la calidad del trazo es muy buena. Hay muchas muestras.

Referencias:

- Specifying colors.
- RGB Colors