Video Vectores

September 18, 2024

1 UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO

1.1 RAUL ALEJANDRO CALDERON HERNANDEZ

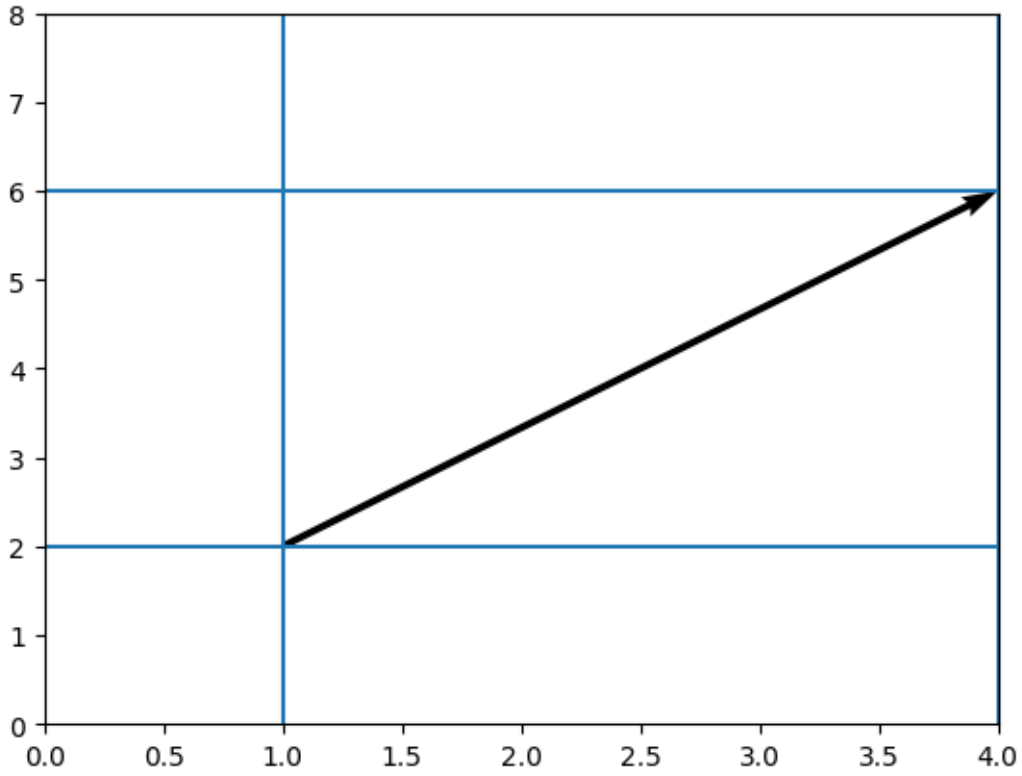
[11]:
```
import numpy as np
import matplotlib.pyplot as plt
import random
```

importamos las librerias necesarias, en este caso el matplotlib y numpy

[7]:
```
plt.quiver(1,2,3,4,scale_units = "xy", angles = "xy", scale = 1 )
plt.xlim(0,4)
plt.ylim(0,8)
plt.axvline(x=1)
plt.axhline(y=2)

plt.axvline(x=1+3)
plt.axhline(y=2+4)
```

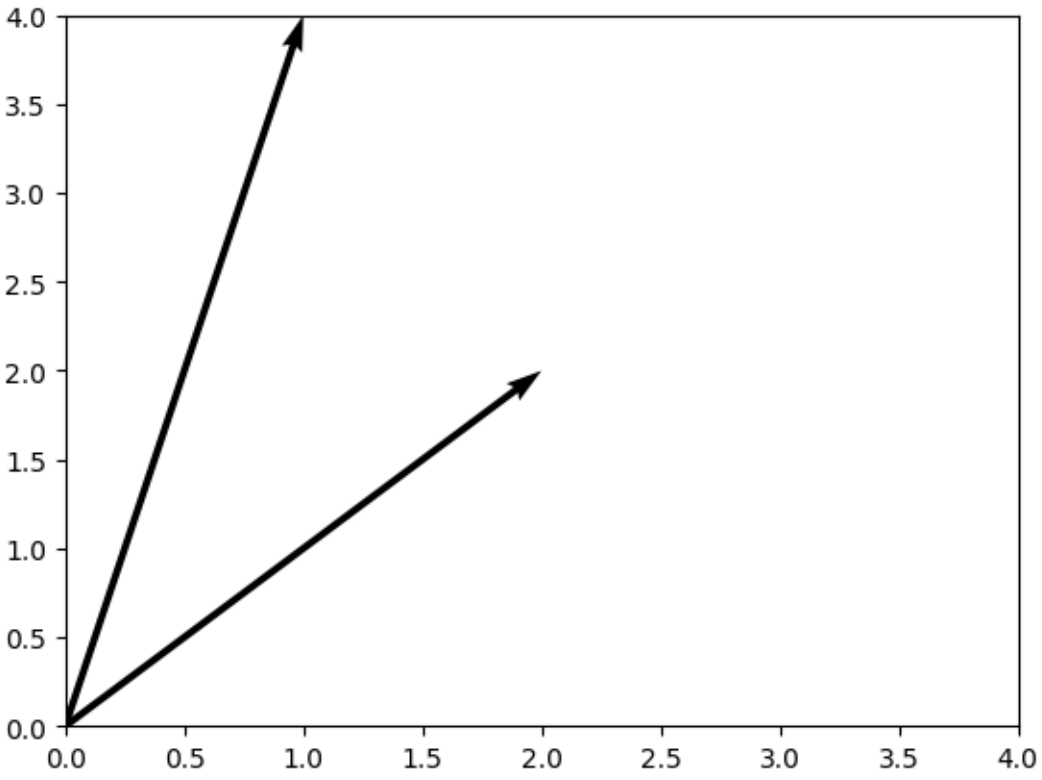
[7]:
```
<matplotlib.lines.Line2D at 0x2df3dc892d0>
```



visualizamos el plano hecho con las escalas unitarias xy y de limites para x de 4 y para y de 8 despues graficamos lineas dependiendo el valor de x y de y en este caso seria de 6 y 2 para y y de 1 para x y por ultimo usamos los vectores

[9]:
```
plt.quiver([0,0],[0,0],[1,2],[4,2],scale_units = "xy", angles = "xy", scale = 1␣
↪)
plt.xlim(0,4)
plt.ylim(0,4)
```

[9]:
```
(0.0, 4.0)
```



podemos observar que podemos poner una lista de arreglos bidimensionales para graficar vectores

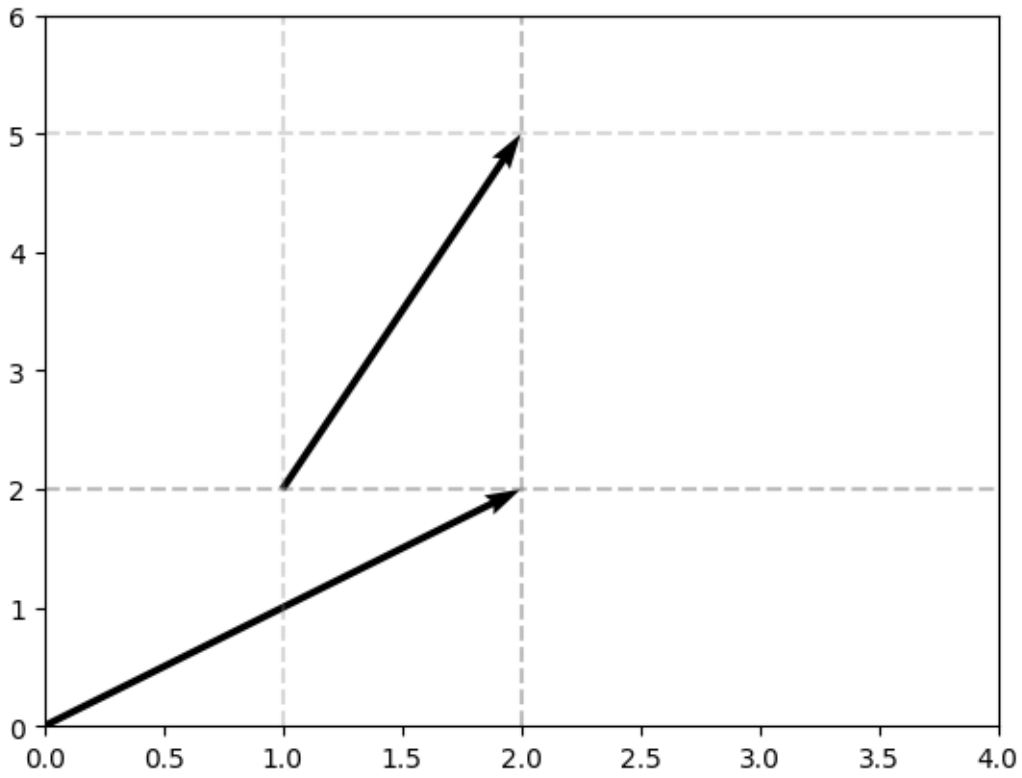
[15]:
```
plt.quiver([1,0],[2,0],[1,2],[3,2],scale_units = "xy", angles = "xy", scale = 1␣
↪)
plt.xlim(0,4)
plt.ylim(0,6)

plt.axvline(x=1,color = "gray", linestyle = "--", alpha = 0.3)
plt.axhline(y=2,color = "gray", linestyle = "--", alpha = 0.3)

plt.axvline(x=2,color = "gray", linestyle = "--", alpha = 0.3)
plt.axhline(y=2,color = "gray", linestyle = "--", alpha = 0.3)

plt.axvline(x=2,color = "gray", linestyle = "--", alpha = 0.3)
plt.axhline(y=5,color = "gray", linestyle = "--", alpha = 0.3)
```

[15]:
```
<matplotlib.lines.Line2D at 0x20b56603150>
```



utilizamos las lineas para graficar donde empieza y donde termina cierto vector

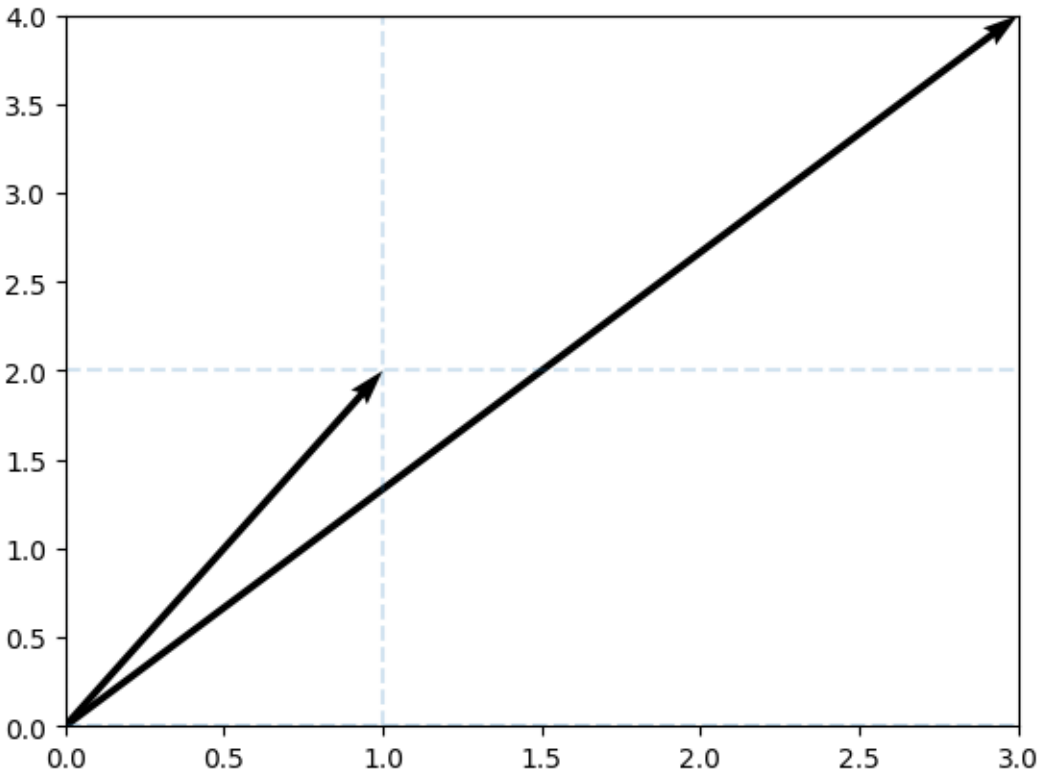
[18]:
```
v1 = np.array([1,2])
v2 = np.array([3,4])

plt.quiver([0,0],[0,0],[v1[0],v2[0]],[v1[1],v2[1]],angles = "xy",␣
 ↪scale_units="xy",scale = 1)
plt.xlim(0,max(v1[0],v2[0]))
plt.ylim(0,max(v1[1],v2[1]))

plt.axvline(x=1, linestyle = "--", alpha = 0.2)
plt.axhline(y=2, linestyle = "--", alpha = 0.2)


plt.axvline(x=0, linestyle = "--", alpha = 0.3)
plt.axhline(y=0, linestyle = "--", alpha = 0.3)
```

[18]: <matplotlib.lines.Line2D at 0x20b56757690>

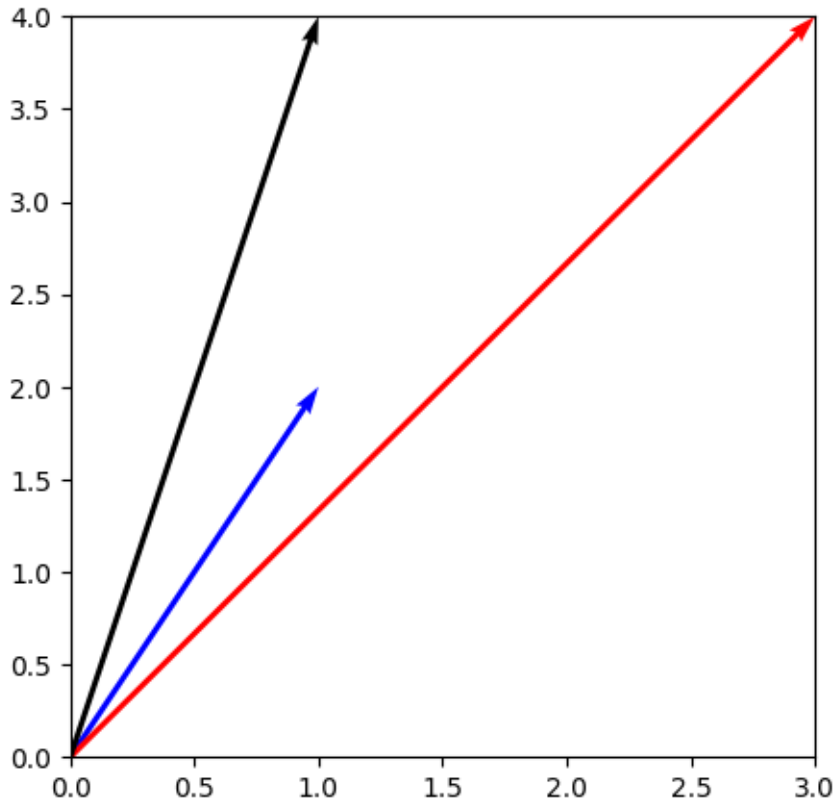


[4]:
```
def grafvecs(vecs,cols):
    plt.figure(figsize=(5,5))
    for i in range(len(vecs)):
        vec = vecs[i]
        plt.quiver(0,0,vec[0],vec[1],color = cols[i],
                   angles = "xy", scale_units="xy",scale=1)
```

creamos un metodo para graficar vectores, con colores, con ayuda de un for

[6]:
```
v1 = np.array([1,2])
v2 = np.array([3,4])
v3 = np.array([1,4])
grafvecs([v1,v2,v3],['blue','red','black'])
plt.xlim(0,max(v1[0],v2[0]))
plt.ylim(0,max(v1[1],v2[1]))
```

[6]:
```
(0.0, 4.0)
```



visualizacion final de como se usa el metodo

2 EJEMPLOS DE VECTORES

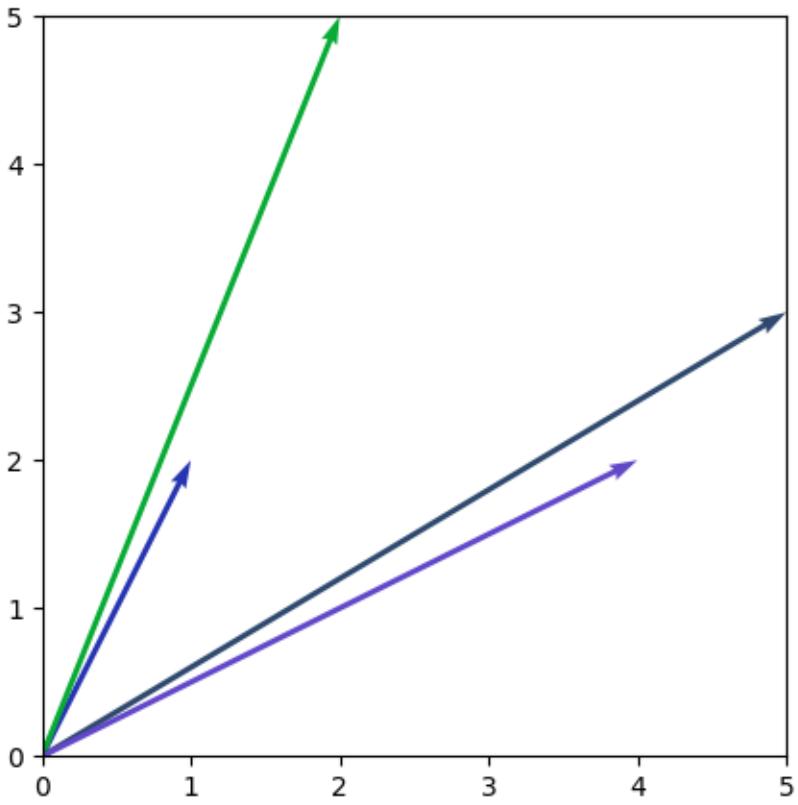
podemos modificar un poco el codigo, para que en vez de que lo haga uno a uno pueda recibir una lista y realizar los vectores, por ejemplo

[1]:
```
def grafvecs(vecs,cols):
    plt.figure(figsize=(5,5))
    for i in range(len(vecs)):
        vec = vecs[i]
        plt.quiver(0,0,vec[0],vec[1],color = cols[i],
                   angles = "xy", scale_units="xy",scale=1)
```

[13]:
```
lista = [[1,2],[2,5],[5,3],[4,2]]
colores = []
for i in range(len(lista)):
    r = random.randint(0, 255)  # Rojo
    g = random.randint(0, 255)  # Verde
    b = random.randint(0, 255)  # Azul
```

```
    # Convertir los valores RGB a hexadecimal y devolver el color en formato hex
    colores.append( '#{:02x}{:02x}{:02x}'.format(r, g, b))
grafvecs(lista,colores)
plt.xlim(0, max([i[0] for i in lista]))
plt.ylim(0, max([i[1] for i in lista]))
```

[13]: (0.0, 5.0)



Claramente, si iniciamos un nuevo vector, en la “lista” pero en realidad es una lista de vectores, se inicializara un for, con los colores, que no es tan relevante, pero dentro del for, encontramos los RGB, valores que despues se volveran hexadecimales, y despues se llamara al metodo para graficar vectores, posteriormente, hacemos como un lambda de for, que nos permite realizar el maximo numero de los valores especificos de los vectores en ese caso X y Y para despues modificar el plano para que tengan los mismos parametros del mayor vector posible en las direcciones de X y Y

[ ]: