Con vectores, nos dan un punto, y el origen es 0,0

POR FUNCION

import numpy as np

import matplotlib.pyplot as plt

def f(m,x,b):

   return m\*x+b

m = 10

b =5

#array 100 valores reales del -10 al 10 igual distancia entre ellos

x = np.linspace(-10,10, num=100)

y =f(m,x,b)

fig, ax = plt.subplots()

ax.plot(x,y)

ax.grid()

ax.axhline(y=0, color='r')

ax.axvline(x=0, color='r')

Chart, line chart

Description automatically generated

GRAFICAR 2 VECTORES PUNTOS EN 2D

import numpy as np

import matplotlib.pyplot as plt

def graficarVectores(vecs, cols, alpha=1):

    plt.figure()

    plt.axvline(x=0, color="grey", zorder=0)

    plt.axhline(y=0, color="grey", zorder=0)

    for i in range(len(vecs)):

        x=np.concatenate([[0,0], vecs[i]])

        plt.quiver([x[0]],

                   [x[1]],

                   [x[2]],

                   [x[3]],

                   angles='xy', scale\_units='xy', scale=1,

                     color=cols[i], alpha=alpha)

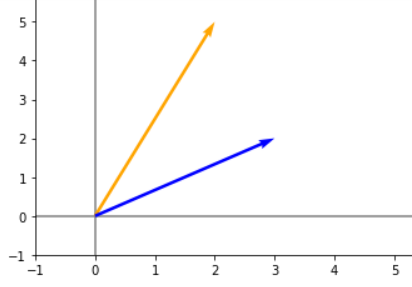
v1=np.array([2,5])

v2=np.array([3,2])

graficarVectores([v1,v2], ['orange', 'blue'])

plt.xlim(-1,6)

plt.ylim(-1,6)



GRAFICAR 3 VECTORES 3VECORES 2 D

v1 = np.array([2,5])

v2 = np.array([3,2])

v3 = 2\*v1 + 3\*v2

graficarVectores([v1,v2,v3], ["orange","blue","red"])

plt.xlim(-1,8)

plt.ylim(-1,8)

Chart, line chart

Description automatically generated

PRESENTER AREA

Text

Description automatically generated

Chart

Description automatically generated

3 D

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

import numpy as np

vectors=np.array( [ [0,0,1,1,-2,0],

[0,0,2,1,1,0],

[0,0,3,2,1,0],

[0,0,4,0.5,0.7,0]])

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

for vector in vectors:

    v = np.array([vector[3],vector[4],vector[5]])

    vlength=np.linalg.norm(v)

    ax.quiver(vector[0],vector[1],vector[2],vector[3],vector[4],vector[5],

            pivot='tail',length=vlength,arrow\_length\_ratio=0.3/vlength)

ax.set\_xlim([-4,4])

ax.set\_ylim([-4,4])

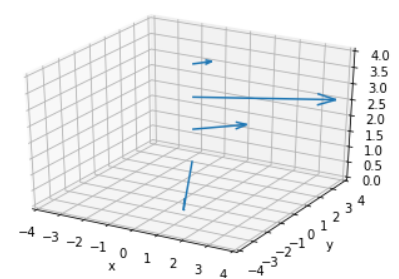
ax.set\_zlim([0,4])

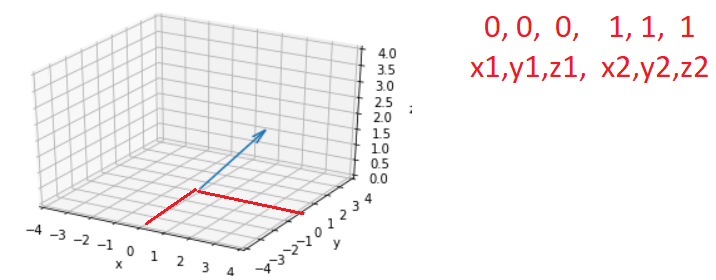
ax.set\_xlabel('x')

ax.set\_ylabel('y')

ax.set\_zlabel('z')

plt.show()





import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

import numpy as np

vectors=np.array( [ [0,0,0,0,-1,0],

                    [0,0,0,-1,0,0],

                    [0,0,0,0,0,1],

                    [0,0,0,0,0,-1]])

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

i=0

for vector in vectors:

    v = np.array([vector[3],vector[4],vector[5]])

    vlength=np.linalg.norm(v)

    colores = 'red'

    if i==1:

        colores = 'blue'

    if i==2:

        colores = 'orange'

    if i==3:

        colores = 'black'

    i=i+1

    ax.quiver(vector[0],vector[1],vector[2],vector[3],vector[4],vector[5],

            pivot='tail',length=vlength,arrow\_length\_ratio=0.3/vlength,

            color=colores )

ax.set\_xlim([-1,1])

ax.set\_ylim([-1,1])

ax.set\_zlim([-1,1])

ax.set\_xlabel('x')

ax.set\_ylabel('y')

ax.set\_zlabel('z')

plt.show()

