Calcula el ángulo (en grados) entre los vectores a=[0,-1,0] y b=[-1,0,0].

Construye un vector ortogonal a ambos vectores.

a

-90° y [0,0,-1]

b

90° y [0,0,-1]

c

1090° y [0,0,1]

d

π° y [0,0,-1]

e

0° y [0,0,1]

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'

tit = ‘v1 0,-1,0’

    if i==1:

        colores = 'blue'

tit=’v2 -1,0,0’

    if i==2:

        colores = 'orange'

tit = ‘v3 0,0,1’

    if i==3:

        colores = 'black'

tit = ‘v4 0,0-1’

    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, label=tit )

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()

Chart, line chart, scatter chart

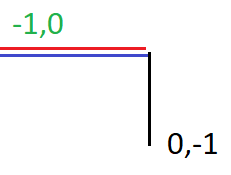
Description automatically generated

Construye un vector ortogonal a ambos vectores.

a

90° y [0,0,-1]

El vector negro (0,0,-1) hace un Angulo -90 con ambos



-90° y [0,0,-1] X

Determina el coeficiente de correlación de las dos siguientes series de tiempo:

a = [ 1.95549103, 2.01534324, 1.57476236, 0.41230402, -1.64797285, -1.88712578,

-1.69938709, -1.14554822, 0.74721043, 1.40194224]

b = [-2.08024322, -0.57860175, 0.4719788, 1.40672063, 2.39433809, 1.85156671,

0.86200697, -0.9201432, -2.30401734, -2.16378775]

a = [ 1.95549103,  2.01534324,  1.57476236,  0.41230402, -1.64797285, -1.88712578,

 -1.69938709, -1.14554822, 0.74721043, 1.40194224]

b = [-2.08024322, -0.57860175,  0.4719788, 1.40672063, 2.39433809, 1.85156671,

  0.86200697, -0.9201432, -2.30401734,  -2.16378775]

x = np.array(a)

y = np.array(b)

my\_rho = np.corrcoef(x, y)

print(my\_rho)

-0.64620219

Regresando a el set de datos de [cáncer de mama](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html" \l "sklearn.datasets.load_breast_cancer" \t "_blank).

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load\_breast\_cancer.html#sklearn.datasets.load\_breast\_cancer

**import** numpy **as** np

**from** sklearn.datasets **import** load\_breast\_cancer

**dataSet** = load\_breast\_cancer()

**data** = dataSet.**data**

Cuál de los siguientes **NO** es un atributo que se mide en el experimento.

**autoimagen**

Cuántas mediciones tenemos y cuál es la dimensión del espacio, i.e

¿Cuál es el tamaño del vectory cuál es la dimensión de cada?

Size(data) = 569 size(data i) = 30

Usando el algoritmo de K-Means que diseñamos en el curso en Python y usando como dos vectores característicos iniciales los dos primeros elementos de, cuenta la cantidad de 1 y 2 que obtienes en la clasificación.

Esa clasificación que obtienes es un modelo simplificado para detección de cáncer.

212(M),357(B)

a

Elementos

Cat 1 = 10

Cat 2 = 5

Cat 1 = 438

Cat 2 = 5

443

Cat 1 = 438

Cat 2 = 1

439

Cat 1 = 438

Cat 2 = 131

569

Cat 1 = 45

Cat 2 = 100

145

Partiendo que,

Cuál es la fórmula para calcular.

Text

Description automatically generated



A+B = 8.4, 3, 8, 9

A-B = 5.6, 3, -6, 3

A.dot(B) = 7\*1.4 + 0 + 7 + 18 = 34.8