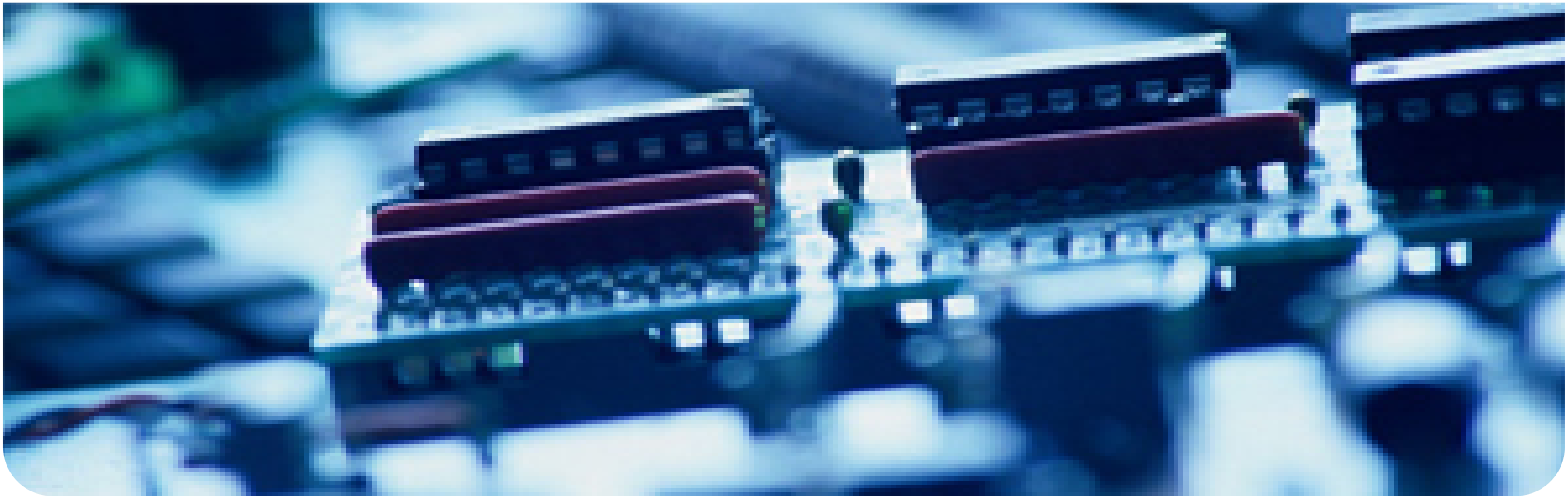
数据挖掘和大数据分析



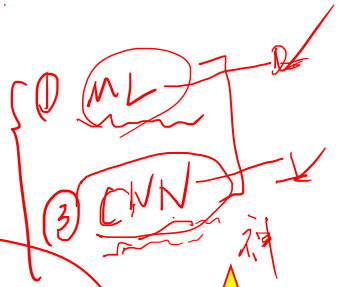
Outline

① Review （Assignments & Data Cleaning ）

② Data Analysis & Python

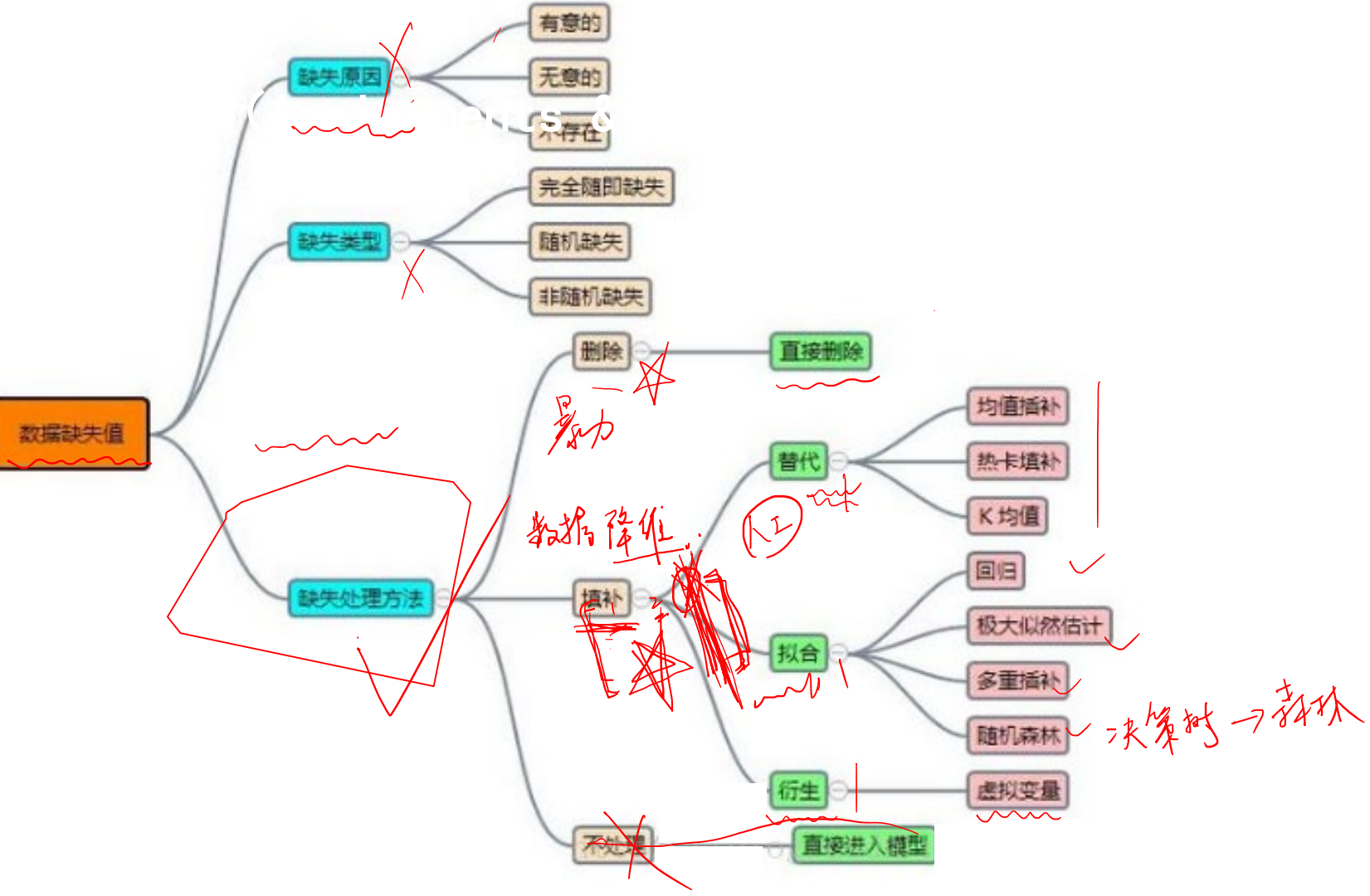
③ KNN Algorithm

④ Euclidean Geometric Distance



DATA ANALYTICS:

DATA MINING AND BIG DATA


数据缺失值
缺失原因
有意的
无意的
不存在
缺失类型
完全随即缺失
随机缺失
非随机缺失
缺失处理方法
删除
直接删除
替代
均值插补
热卡填补
K均值
填补
拟合
回归
极大似然估计
多重插补
随机森林
衍生
虚拟变量
不处理
直接进入模型

Review （Assignments & Data Cleaning ）

【2】 对下图的数据采用删除和填补两种方法进行清洗。

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	NA	0.2	setosa
4.7	3.2	NA	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.1	3.8	NA	0.2	setosa
4.6	3.2	1.4	0.2	setosa
5.3	NA	1.5	0.2	setosa
5	3.3	1.4	0.2	setosa
7	3.2	4.7	1.4	versicolor
6.4	3.2	NA	1.5	versicolor
6.3	3.3	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
NA	3	NA	2.1	virginica
6.3	2.9	5.6	1.8	virginica
5.9	3	NA	1.8	virginica

```
data=pd.read_csv("test4.csv")

print(data.dropna(axis=1))    #这一列如果有缺失,删除这一列

print(data.dropna(axis=0))    #这一行如果有缺失,删除这一行


print(data.fillna(0))    #缺失的地方补 0

print(data.fillna(method='pad')) #缺的地方用同列前一行的值填充

print(data.fillna(data.mean()))    #缺的地方用同列的均值填充

print(data.fillna(data.median())) #缺的地方用同列的中位数填充

print(data.fillna(data.mode()))    #缺的地方用同列的众数填充
```

投票　最多可选1项

设置

Do you finish it in 5 minutes?

A Yes

B No

提交

A	B	C	D
TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5			10.4
151.5	41.3	58.5	18.5
180.8			12.9
8.7	48.9	75	7.2

```
#这一列如果有缺失,删除这一列

#这一行如果有缺失,删除这一行
```

```
#缺失的地方补 0

od='pad')) #缺的地方用同列前一行的值填充

nean()))   #缺的地方用同列的均值填充

nedian())) #缺的地方用同列的中位数填充

node()))   #缺的地方用同列的众数填充
```

Review （Assignments & Data Cleaning ）

3 对【2】题数据采用回归方法填补。

```
import numpy as np
import pandas
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer

data = pandas.read_csv("test4.csv")
print(data)

imp = IterativeImputer(max_iter=10, random_state=0)#回归填补法
imp.fit(data)
v = np.round(imp.transform(data))
print(v)
```

Review （Assignments & Data Cleaning ）

3 对【2】题数据采用回归方法填补。

【3】 对【2】题数据采用回归方法填补。

采用一元线性回归，由于只有 Sepal.Length、Sepal.Width、Petal.Length 三列存在缺失值，将这三列单独拿出来与最后一列各自组成一个 csv 文件，由于最后一列是英文名将其按种类分别映射成数字 0,1,2，也就是 setosa 对应 0、versicolor 对应 0、virginica 对应 1。再分三次回归填补。相当于构建三条回归直线。

整体思路都是先删除有缺失值的行，然后再构建线性回归模型，再预测缺失值。

Review （Assignments & Data Cleaning ）

3 对【2】题数据采用回归方法填补。

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	NA	0.2	setosa
4.7	3.2	NA	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.1	3.8	NA	0.2	setosa
4.6	3.2	1.4	0.2	setosa
5.3	NA	1.5	0.2	setosa
5	3.3	1.4	0.2	setosa
7	3.2	4.7	1.4	versicolor
6.4	3.2	NA	1.5	versicolor
6.3	3.3	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
NA	3	NA	2.1	virginica
6.3	2.9	5.6	1.8	virginica
5.9	3	NA	1.8	virginica

#Sepal.Length 列的填补

```
data1=pd.read_csv("test4-1.csv")
data1=data1.dropna(axis=0)
regr1=linear_model.LinearRegression()
regr1.fit(data1['species'].values.reshape(-1,1),data1['Sepal.Length'])
print(regr1.predict([[2]]))
```

结果如下：

[6.32894737]

Data Analysis & Python

import numpy

import pandas

Data Format

import matplotlib

import matplotlib.pyplot

import Seaborn

Data Visualization

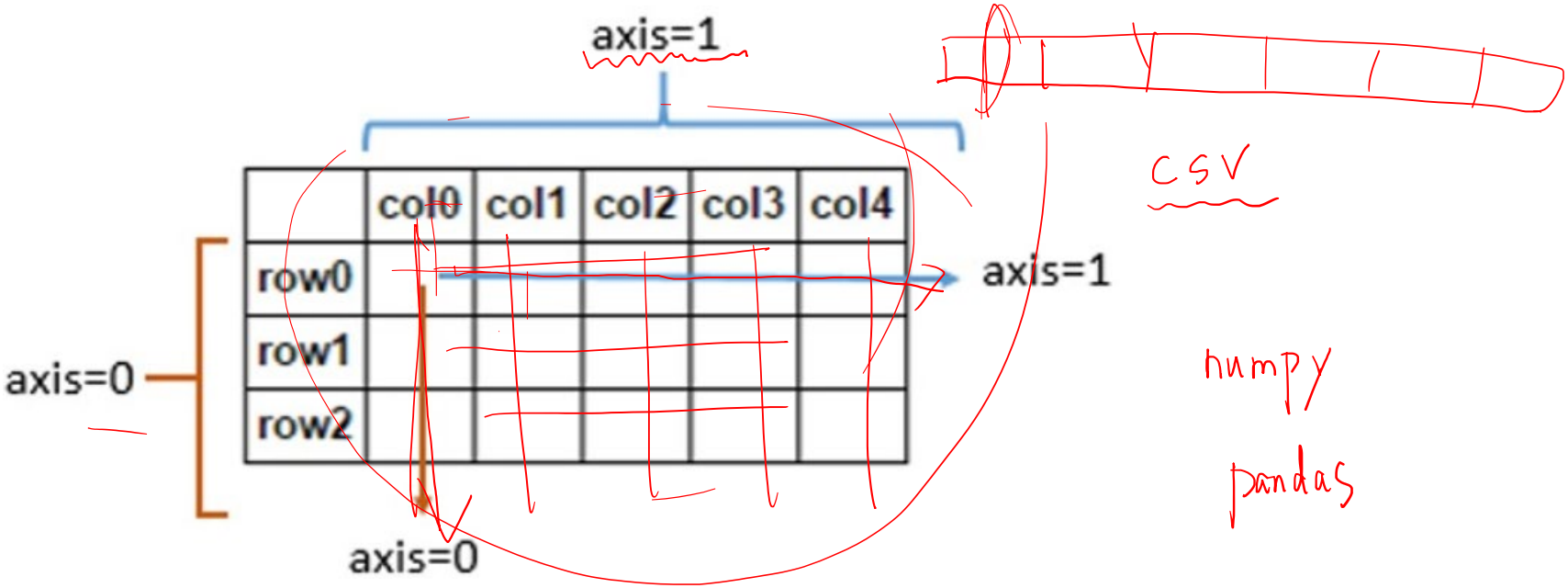
import sklearn

Machine Learning

import scipy

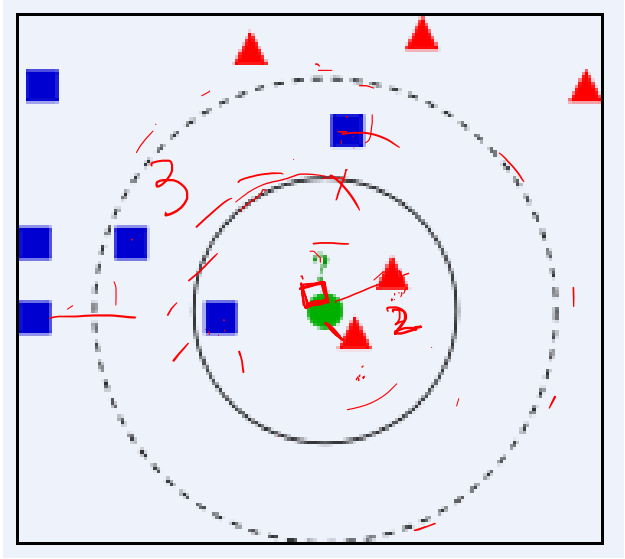
[e.g] 插值，积分，优化，……

Data Analysis & Python



KNN Algorithm

KNN (K-NearestNeighbor)



Reading KNN code

欧式距离： $d(x,y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2}$ ，

曼哈顿距离： $d(x,y)=\sqrt{\sum_{k=1}^{n}|x_k-y_k|}$

投票　最多可选1项

设置

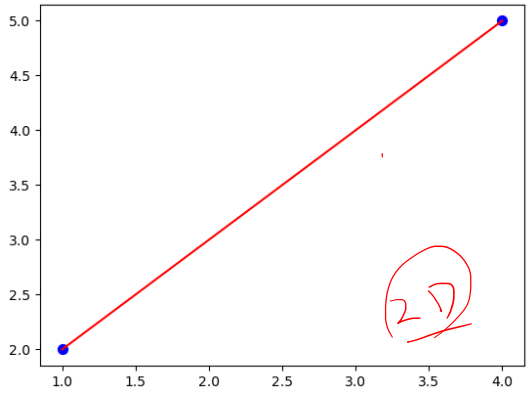
How do you understand KNN and Kmeans?

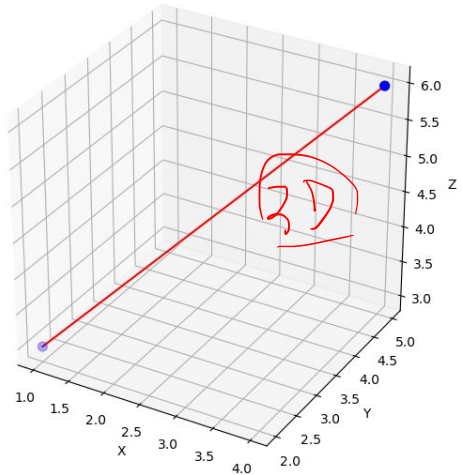
A　KNN = = K'means

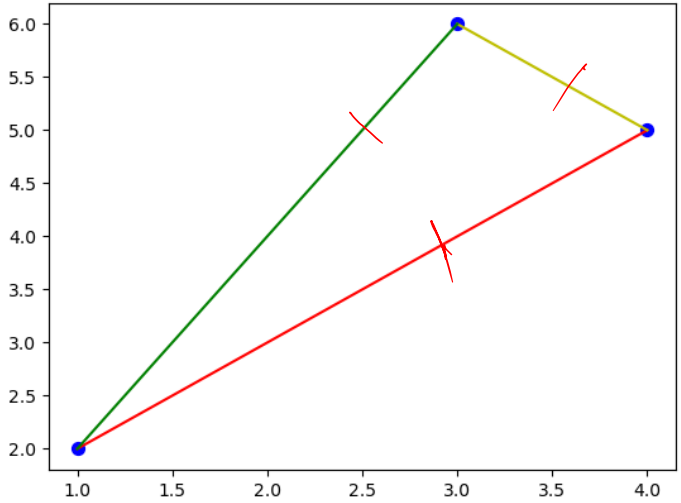
B　KNN ! = K'means

提交

Lab 5: Euclidean Geometric Distance

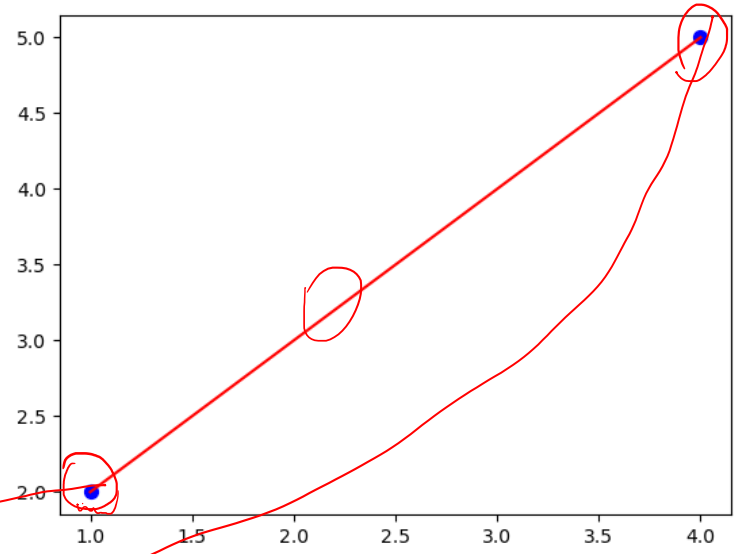






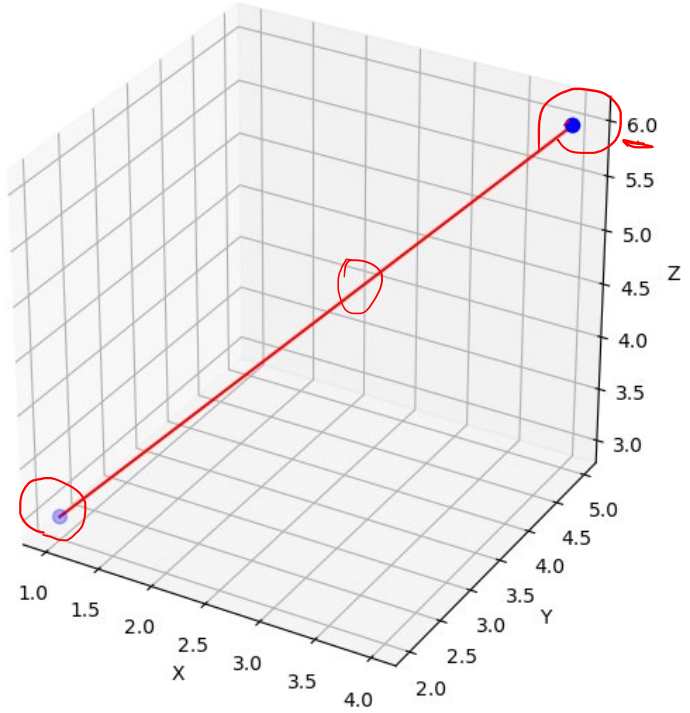
Lab 5: Euclidean Geometric Distance

```
import numpy as np
from matplotlib import pyplot as plt
import math
def eucliDist(A,B):
    return math.sqrt(sum([(a - b)**2 for (a,b) in zip(A,B)]))
coords1 = [1, 2]
coords2 = [4, 5]
print(eucliDist(coords1,coords2))
plt.scatter((coords1[0], coords2[0]),(coords1[1], coords2[1]),color="b", s=50)
plt.plot((coords1[0], coords2[0]),(coords1[1], coords2[1]),color="r")
plt.show()
```



Lab 5: Euclidean Geometric Distance

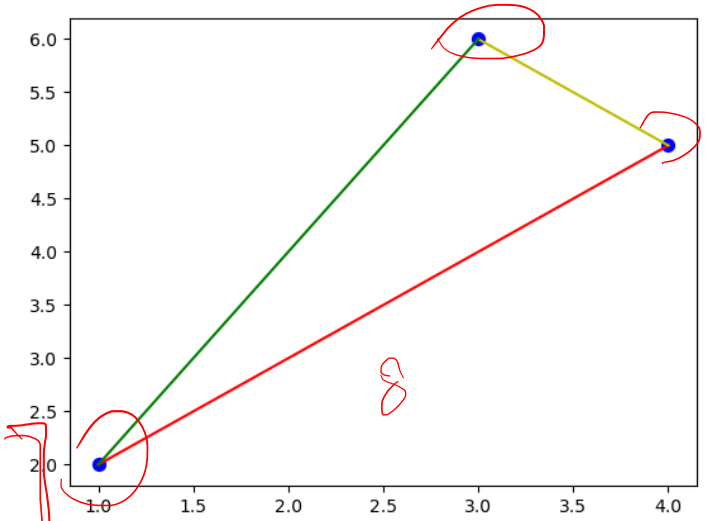
```
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.mplot3d import proj3d
coords1 = [1, 2, 3]
coords2 = [4, 5, 6]
fig = plt.figure(figsize = (7,7))
ax = fig.add_subplot(111, projection='3d')
ax.scatter((coords1[0], coords2[0]),(coords1[1],
coords2[1]),(coords1[2], coords2[2]),color="b", s=50)
ax.plot((coords1[0], coords2[0]),(coords1[1],
coords2[1]),(coords1[2], coords2[2]),color="r")
```



Lab 5: Euclidean Geometric Distance

```
coords3 = [3, 6]
d1 = eucliDist(coords1,coords2)
print(d1)
d2 = eucliDist(coords2,coords3)
print(d2)
d3 = eucliDist(coords1,coords3)
print(d3)
```

```
plt.scatter((coords1[0], coords2[0],coords3[0]),(coords1[1],
coords2[1],coords3[1]),color="b", s=50)
plt.plot((coords1[0], coords2[0]),(coords1[1], coords2[1]),color="r")
plt.plot((coords2[0], coords3[0]),(coords2[1], coords3[1]),color="y")
plt.plot((coords1[0], coords3[0]),(coords1[1], coords3[1]),color="g")
```



贵在坚持！