**- 2주차 과제 (전수경) -**

**1. numpy의 range와 meshgird함수를 활용해서 아래 문제를 해결해주세요**

**0 <= x <= 5, x는 정수**

**0 <= y <= 5, y는 정수**

**위 조건을 만족하는 모든 x,y 좌표에 대해 plot에 점을 찍어주세요**

- 필요한 라이브러리 생성

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

→ from matplotlib import pyplot as plt 도 가능

- xvalues와 yvalues를 range이용하여 생성

xvalues = list(range(0,6,1))

yvalues = list(range(0,6,1))

\* meshgrid 함수란?

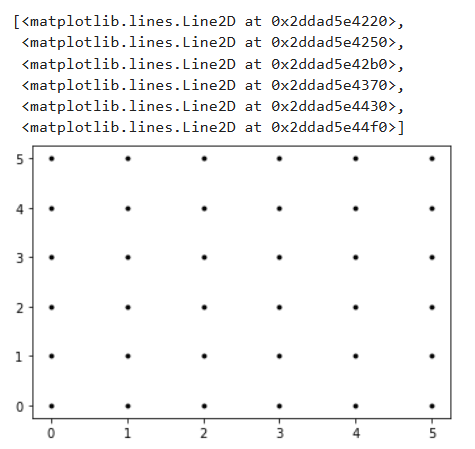
numpy 에 속한 함수로 2차원, 3차원 그래프를 그릴 때 자주 사용

주어진 1차원 배열을 바탕으로 좌표 행렬을 생성하며 각 요소를 조합하여, 각각의 축에 대한 좌표 행렬을 반환함 (격자점)

x, y = np.meshgrid(xvalues, yvalues)

- 실제 그래프 그리기

plt.plot(x,y, marker='.', color='k', linestyle='none')



**2-1 ) pandas로, 아래와 같은 데이터를 만든 후에 to\_csv를 활용해서 데이터를 csv형태로 저장합니다.**

**\* header 형식 : 학번 점수**

**\* 학번은 1 ~ 100000까지 sequential하게 하고 점수는 0~100의 값을 랜덤으로 작성합니다.**

- 학번

student\_ids = np.arange(1, 1000001)

- 점수

scores = np.random.unifrom(0, 101, size=1000000)

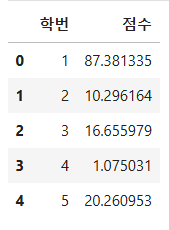
- 데이터프레임 생성

df = pd.DataFrame({'학번': student\_ids, '점수': scores})

- csv형태로 저장

df.to\_csv('export\_dataframe.csv', index=False, header=True, encoding='utf-8')

- head(5)



**2-2) read\_csv를 통해 해당 데이터를 읽어옵니다**

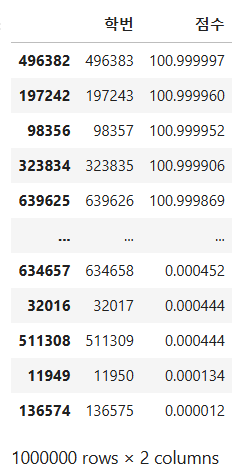
**2-3) 점수를 기준으로 내림차순 정렬 합니다**

- read\_csv 이용해 데이터 읽어오기

df2 = pd.read\_csv("export\_dataframe.csv")

- 점수 기준으로 내림차순 정렬

df3 = df2.sort\_values(by='점수', ascending=False)



**2-4) 아래 값들을 구합니다.**

**\* 전체 학생의 평균,분산**

**\* 상위 5%학생과 하위 5%학생의 평균 ====> 제거**

- 평균, 분산

print("평균 = ", df3['점수'].mean())

print("분산 = ", df3['점수'].var())



- 상위 5% 학생과 하위 5% 학생의 평균 제거

top = int(100000 \* 0.05)

bottom = int(100000 \* 0.95)

df4 = df3.iloc[top:bottom]

print("평균 = ", df4['점수'].mean())

print("분산 = ", df4['점수'].var())



**2-5) grade라는 column을 하나 생성해서 1~5로 등급을 표시합니다.**

**\* 1등급 : 상위 5%**

**\* 2등급 : 상위 6~13%**

**\* 3등급 : 상위 14~24%**

**\* 4등급 : 상위 25~72%**

**\* 5등급 : 상위 73~100%**

- 등급 부여 함수 정의

def getGrade(x,gradeList):

if(x['점수'] >= gradeList[0]):

return "1";

elif(x['점수'] >= gradeList[1]):

return "2";

elif(x['점수'] >= gradeList[2]):

return "3";

elif(x['점수'] >= gradeList[3]):

return "4";

elif(x['점수']>= gradeList[4]):

return "5";

- 등급의 경계값 계산

first\_cut = df3['점수'].iloc[int(1000000\*0.05)]

second\_cut = df3['점수'].iloc[int(1000000\*0.13)]

third\_cut = df3['점수'].iloc[int(1000000\*0.24)]

forth\_cut = df3['점수'].iloc[int(1000000\*0.72)]

fifth\_cut = df3['점수'].iloc[999999]

- 경계값 배열 생성

gradecut\_list = np.array([first\_cut,second\_cut,third\_cut,forth\_cut,fifth\_cut])

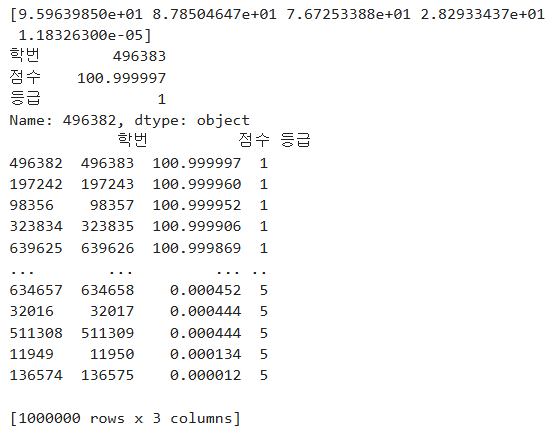
- 데이터 확인 및 등급 부여

print(gradecut\_list)

print(df3.iloc[0])

df3['등급'] = df3.apply(lambda x: getGrade(x,gradecut\_list),axis='columns')

print(df3)



**2-6) 각 등급 인원을 gropuby합니다.**

**2-7) 2-6)에서 저장한 값을 plot으로 출력합니다.**

df5 = df3.groupby('등급')

print("\nmean\n",df5['점수'].mean().sort\_values(ascending=False))

print("\nvar\n",df5['점수'].var().sort\_values(ascending=False))

print("\nstd\n",df5['점수'].std().sort\_values(ascending=False))

print("\nstd\n",df5['점수'].size().sort\_values(ascending=False))

df5['점수'].size().sort\_values(0,ascending=True).plot()

