

Homework Assignment Hw1

보고서 및 논문 윤리 서약

1. 나는 보고서 및 논문의 내용을 조작하지 않겠습니다.
2. 나는 다른 사람의 보고서 및 논문의 내용을 내 것처럼 무단으로 복사하지 않겠습니다.
3. 나는 다른 사람의 보고서 및 논문의 내용을 참고하거나 인용할 시 참고 및 인용 형식을 갖추고 출처를 반드시 밝히겠습니다.
4. 나는 보고서 및 논문을 대신하여 작성하도록 청탁하지도 청탁받지도 않겠습니다.

나는 보고서 및 논문 작성 시 위법 행위를 하지 않고, 명지인으로서 또한 공학인으로
서 나의 양심과 명예를 지킬 것을 약속합니다.



학 과 : 융합소프트웨어학부 데이터테크놀로지전공

과 목 : 인공지능

담당교수 : 전종훈

강좌 번호: 6019

학 번 : 60182196

이 름 : 이동혁 (서명)

1.

A.

```
# Q1.a
a = {'dep': ['cs', 'cs', 'dt', 'dcd', 'dt', 'as', 'as', 'ice', 'ice', 'ice'],
      'sno': ['0001', '0002', '0003', '0004', '0005', '0006', '0007', '0008', '0009', '0010'],
      'gpa': [2.10, 3.50, 4.50, 2.70, 3.00, 3.15, 4.00, 0.00, 3.25, 3.70]}
print(a)

{'dep': ['cs', 'cs', 'dt', 'dcd', 'dt', 'as', 'as', 'ice', 'ice', 'ice'], 'sno': ['0001', '0002', '0003', '0004', '0005', '0006', '0007', '0008', '0009', '0010'], 'gpa': [2.1, 3.5, 4.5, 2.7, 3.0, 3.15, 4.0, 0.0, 3.25, 3.7]}
```

B.

```
# Q1.b
import numpy as np

b = np.array([[ 'cs', 'cs', 'dt', 'dcd', 'dt', 'as', 'as', 'ice', 'ice', 'ice'],
               ['0001', '0002', '0003', '0004', '0005', '0006', '0007', '0008', '0009', '0010'],
               [2.10, 3.50, 4.50, 2.70, 3.00, 3.15, 4.00, 0.00, 3.25, 3.70]])
print(b)

[[ 'cs' 'cs' 'dt' 'dcd' 'dt' 'as' 'as' 'ice' 'ice' 'ice']
 ['0001' '0002' '0003' '0004' '0005' '0006' '0007' '0008' '0009' '0010']
 [2.1  3.5  4.5  2.7  3.0  3.15  4.0  0.0  3.25  3.7]]
```

C.

```
# Q1.c
import pandas as pd

c = pd.DataFrame({
    'dep': ['cs', 'cs', 'dt', 'dcd', 'dt', 'as', 'as', 'ice', 'ice', 'ice'],
    'sno': ['0001', '0002', '0003', '0004', '0005', '0006', '0007', '0008', '0009', '0010'],
    'gpa': [2.10, 3.50, 4.50, 2.70, 3.00, 3.15, 4.00, 0.00, 3.25, 3.70]
})
print(c.T)

      0   1   2   3   4   5   6   7   8   9
dep  cs  cs  dt  dcd  dt  as  as  ice  ice  ice
sno 0001 0002 0003 0004 0005 0006 0007 0008 0009 0010
gpa 2.1  3.5  4.5  2.7   3  3.15   4   0  3.25  3.7
```

2.

A.

```
# Q2.a

print(round(sum(a['gpa'])/len(a['gpa']),2))

2.99
```

B.

```
# Q2.b  
  
x = np.array(b[2],dtype = float)  
  
print(np.mean(x))  
print(np.median(x))
```

2.99
3.2

C.

```
# Q2.c  
  
print(round(c['gpa'].mean(),2))  
print(c['gpa'].median())
```

2.99
3.2

3.

```
# Q3  
  
def med(lst):  
    lst.sort()  
    while(True):  
        if(len(lst)==1):  
            return lst[0]  
        elif(len(lst)==2):  
            return (lst[0]+lst[1])/2  
        lst.remove(min(lst))  
        lst.remove(max(lst))  
  
print(med(a['gpa']))
```

3.2

4.

```
# Q4  
  
print(c.sort_values('gpa', ascending = False).head(3).tail(1)['gpa'])
```

9 3.7
Name: gpa, dtype: float64

5.

Q5

```
c.groupby('dep').aggregate(['mean',max,min])
```

dep	gpa		
	mean	max	min
as	3.575000	4.0	3.15
cs	2.800000	3.5	2.10
dcd	2.700000	2.7	2.70
dt	3.750000	4.5	3.00
ice	2.316667	3.7	0.00

6.

Q6

```
d = pd.DataFrame ({ 'dep': ["cs", "cs", "dt", "dt", "dt", "as", "as", "cs", "dt", "as"],  
'sno': ["0001", "0002", "0003", "0004", "0005", "0006", "0007", "0008", "0009", "0010"],  
'gpa': [2.10, 3.50, 4.50, 2.70, 3.00, 3.20, 4.00, 0.00, 3.20, 3.70],  
'year': [2018, 2018, 2019, 2019, 2020, 2020, 2020, 2020, 2021, 2021]})  
  
pd.pivot_table(data=d, values='gpa', index='dep', columns='year')
```

year	2018	2019	2020	2021
dep				
as	NaN	NaN	3.6	3.7
cs	2.8	NaN	0.0	NaN
dt	NaN	3.6	3.0	3.2