

# Homework Assignment Hw2

## 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.

```
# 1-a
import numpy as np

lst = [13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40,
45, 46, 52, 70]

print("mean = " + str(np.mean(lst)))
print("median = " + str(np.median(lst)))
```

```
mean = 29.962962962962962
median = 25.0
```

```
# 1-b
import statistics as st

print(st.mode(lst))
print(1.count(st.mode(lst)))
```

```
25
```

```
4
```

Mode가 4개인 Multimodal이다.

```
# 1-c
print(np.quantile(lst,0.25))
print(np.quantile(lst,0.75))
```

```
20.5
```

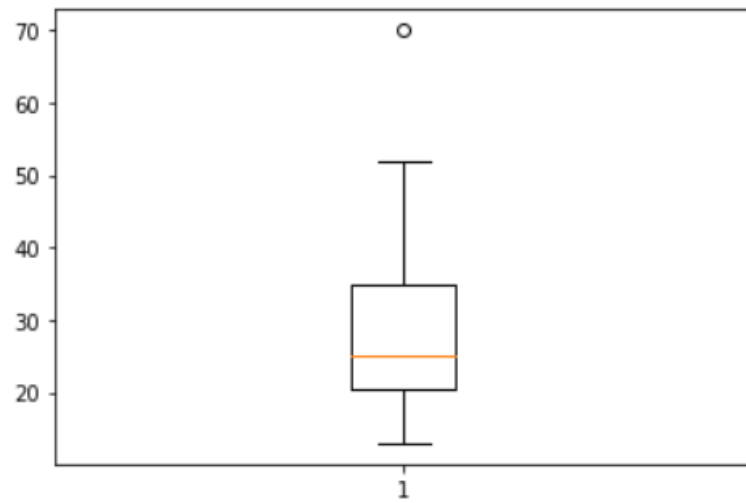
```
35.0
```

```
# 1-d
print(np.percentile(lst, [0, 25, 50, 75, 100]))
```

```
[13. 20.5 25. 35. 70.]
```

```
# 1 - e
import matplotlib.pyplot as plt

plt.boxplot(lst)
plt.show()
```



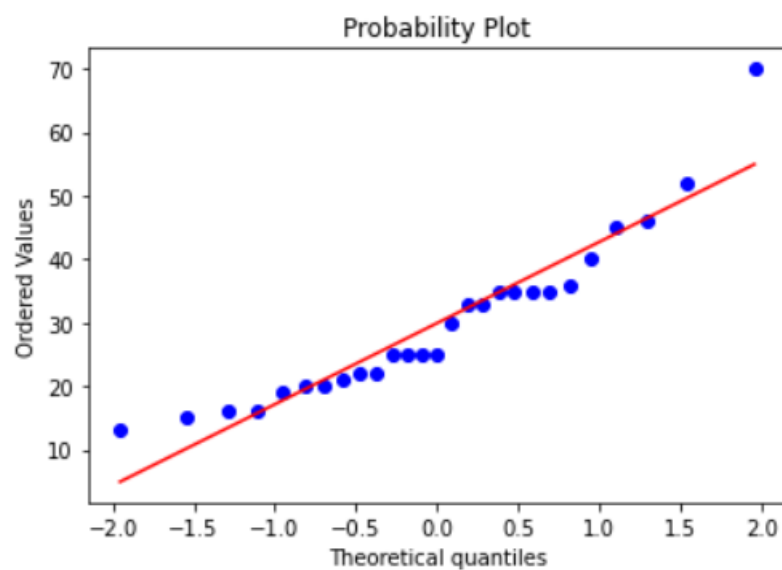
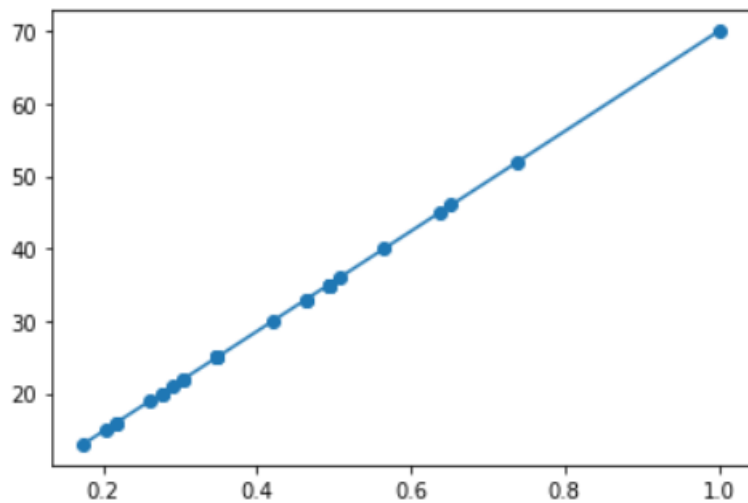
```

# 1 - f
import matplotlib.pyplot as plt
import scipy.stats as sci

# Q plot
x = []
for n in lst:
    x.append((n-1)/(max(lst)-1))
y = lst
plt.plot(x,y)
plt.scatter(x,y)
plt.show()

# Q-Q plot
sci.probplot(lst, dist="norm", plot=plt)

```



x축의 범위가 quantile plot은 [0,1]이지만 q-q plot은 [-2,2]이다. 또 quantile plot의 x축은 분위수를 의미하는 반면 q-q plot의 x축은 누적 확률분포를 의미한다.

2.

```
# 2
import pandas as pd

df = pd.DataFrame.from_dict({'low_range':[1,6,16,21,51,81], 'high_range':[5,15,20,50,80,110], 'frequency':[200,450,300,1500,700,44]})
N = df['frequency'].sum()

# calculating L1
index = abs(df['frequency'].cumsum() - N/2).idxmin()
L1 = df['low_range'][index + 1]
cumsum_before = df['frequency'].cumsum()[index]
freq_medain = df['frequency'][index + 1]
width = df['high_range'][index + 1] - df['low_range'][index + 1] + 1

median = L1 + (N/2 - cumsum_before) / freq_medain * width

print("Approximated median = ", median)

Approximated median = 33.94
```

3.

```

t1 = (22,1,42,10)
t2 = (20,0,36,8)

# 3-a

result = 0
for num in range(len(t1)):
    result += (t1[num]-t2[num])**2
result = result**0.5
print(result)

# 3-b

result = 0
for num in range(len(t1)):
    result += abs(t1[num] - t2[num])
print(result)

# 3-c

lst2 = []
result = 0
for num in range(len(t1)):
    lst2.append(abs(t1[num] - t2[num]))
    result = max(lst2)
print(result)

```

6.708203932499369

11

6

4.

```
# 4
```

```
x1 = (1.5,1.7)
```

```
x2 = (2,1.9)
```

```
x3 = (1.6,1.8)
```

```
x4 = (1.2,1.5)
```

```
x5 = (1.5,1.0)
```

```
X = (x1,x2,x3,x4,x5)
```

```
x = (1.4,1.6)
```

```
print("Euclidean distance")
```

```
for num in range(len(X)):
```

```
    print(((x[0]-X[num][0])**2+(x[1]-X[num][1])**2)**0.5))
```

```
print()
```

```
print("Manhattan distance")
```

```
for num in range(len(X)):
```

```
    print(abs(x[0] - X[num][0]) + abs(x[1] - X[num][1]))
```

```
print()
```

```
print("Supremum distance")
```

```
for num in range(len(X)):
```

```
    print(max(abs(x[0] - X[num][0]),abs(x[1] - X[num][1])))
```

```
print()
```

```
print("Cosine similarity")
```

```
for num in range(len(X)):
```

```
    d1 = (X[num][0]**2+X[num][1]**2)**0.5
```

```
    d2 = (x[0]**2+x[1]**2)**0.5
```

```
    print((X[num][0]*x[0]+X[num][1]*x[1])/(d1*d2))
```

Euclidean distance  
0.14142135623730948  
0.6708203932499369  
0.28284271247461906  
0.22360679774997896  
0.608276253029822

Manhattan distance  
0.19999999999999996  
0.8999999999999999  
0.400000000000000013  
0.30000000000000004  
0.70000000000000002

Supremum distance  
0.10000000000000009  
0.60000000000000001  
0.200000000000000018  
0.19999999999999996  
0.60000000000000001

Cosine similarity  
0.999991391443956  
0.9957522612528874  
0.9999694838187877  
0.9990282349375618  
0.9653633930282662

Euclidean distance : x1, x4, x3, x5, x2

Manhattan distance : x1, x4, x3, x5, x2

Supremum distance : x1, x4, x3, x5,x2

Cosine similarity : x1, x3, x4, x2, x5