Data Mining (CSE542)

Homework 01

ID: ________ Name: _______조원석_______ Date:__'23/03/19______

Task-1: Let X and Y be two random variables, denoting age and weight, respectively. Consider a random sample of size n = 20 from these two variables

X =(69,74,68, 70,72, 67,66, 70, 76, 68, 72, 79, 74, 67,66, 71, 74, 75, 75, 76)

Y =(153,175,155,135, 172, 150,115, 137, 200, 130, 140, 265, 185,112,140,150,165,185,210,220)

(a) Find the mean, median, and mode for X.

```python
import numpy as np
from scipy import stats
def main():
    X = [69,74,68, 70,72, 67,66, 70, 76, 68, 72, 79, 74, 67,66, 71, 74, 75, 75, 76]
    Y = [153,175,155,135, 172, 150,115, 137, 200, 130, 140, 265, 185,112,140,150,165,185,210,220]

    print(f"(a)------------------------------------------------------------")
    x_mean =np.mean(X)
    x_median = np.median(X)
    x_mode = stats.mode(X)[0]
    print(f"mean : {x_mean}") # mean : 71.45
    print(f"median : {x_median}") # median : 71.5
    print(f"mode : {x_mode[0]}") # mode : 74
```

main

```
C:\Users\iqeq1\anaconda3\python.exe "C:\Users\iqeq1\Downloads\code examples-20230314\main.py"
(a)------------------------------------------------------------
mean : 71.45
median : 71.5
mode : 74
```

- mean : 71.45

=> $\text{mean} = \frac{1}{20}\sum_{i=1}^{20} Xi = \frac{1429}{20} = 71.45$

median : 71.5

=> [66, 66, 67, 67, 68, 68, 69, 70, 70, **71, 72**, 72, 74, 74, 74, 75, 75, 76, 76, 79]

[01, 02, 03, 04, 05, 06, 07, 08, 09, **10, 11**, 12, 13, 14, 15, 16, 17, 18, 19, 20]

$10 = \frac{20}{2}, 11 = \frac{20}{2} + 1$

$\frac{71+72}{2}$= 71.5

- mode : 74 (do 3 times)

=> [66, 66, 67, 67, 68, 68, 69, 70, 70, 71, 72, 72, **74, 74, 74**, 75, 75, 76, 76, 79]

(b) What is the variance for Y?

```
import numpy as np
```

```
    print(f"(b)----------------------------------------------------------")
    y_variance = np.var(Y)
    print(f"variance : {y_variance}")
```

main()

main

```
C:\Users\iqeq1\anaconda3\python.exe "C:\Users\iqeq1\Downloads\code examples-20230314\main.py"
(b)----------------------------------------------------------
variance : 1369.2099999999998
```

In [10]:

```
import numpy as np
Y = [153,175,155,135, 172, 150,115, 137, 200, 130, 140, 265, 185,112,140,150,165,185,210,220]
print(np.mean([i**2 for i in Y]) - np.mean(Y)**2)
```

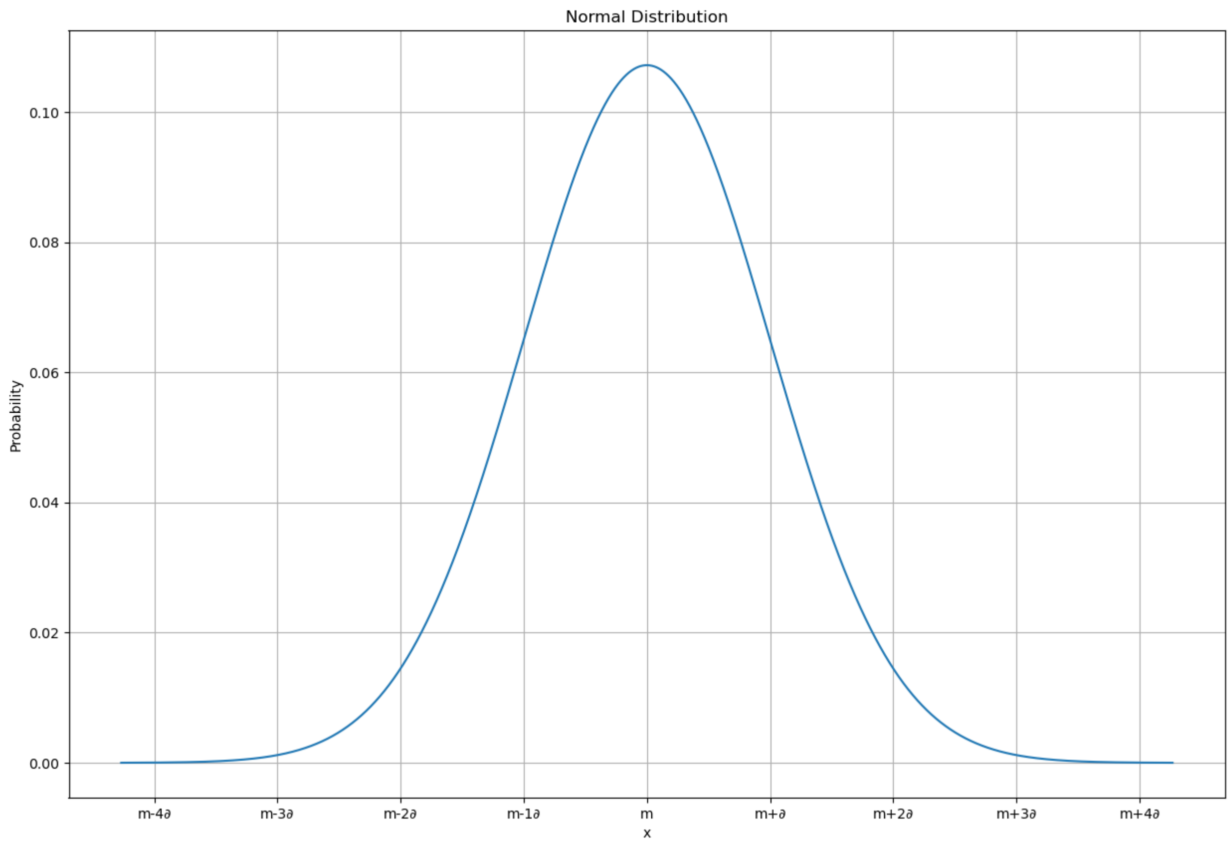
```
1369.2100000000028
```

Y = [153,175,155,135, 172, 150,115, 137, 200, 130, 140, 265, 185,112,140,150,165,185,210,220]

- y_variance ≒ 1369

(c) Plot the normal distribution for X.

```python
print(f"(c)-----------------------------------------------------------")
x_variance = np.var(X)
x_SD = math.sqrt(x_variance)
x = np.arange(x_mean-4*x_SD-1, x_mean + 4*x_SD+1, 0.001)
plt.figure(figsize=(15, 10))
plt.title('Normal Distribution')
plt.xlabel('x')
plt.ylabel('Probability')
plt.xticks(np.arange(x_mean-4*x_SD, x_mean + 4*x_SD+0.001, x_SD), labels=['m-4∂','m-3∂','m-2∂','m-1∂','m','m+∂','m+2∂','m+3∂','m+4∂'])
plt.grid()
plt.plot(x, stats.norm.pdf(x, loc=x_mean, scale=x_SD))
plt.show()
```

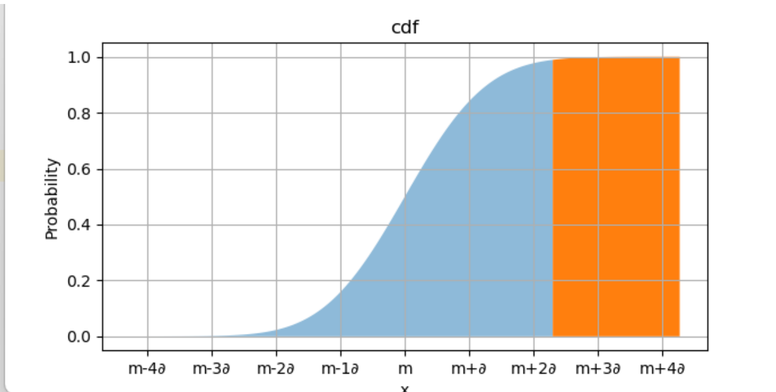



```
(c)-----------------------------------------------------------
x_means: 71.45
x_SD: 3.7212229172679243
```

m=x_means, ∂=x_SD

(d) What is the probability of observing an age of 80 or higher?

- Calculate by z-score method and graph




```
--")
_SD),
', 'm+3∂', 'm+4∂'])
plt.fill_between(x[x <= 80], cdf[x <= 80], alpha=0.5)
plt.fill_between(x[x >= 80], cdf[x >= 80])
print(f"probability of observing an age of 80 or higher: {stats.norm.cdf(1 - (80 - x_mean) / x_SD)}")
```

in()

main ×

```
C:\Users\iqeq1\anaconda3\python.exe "C:\Users\iqeq1\Downloads\code examples-20230314\main.py"
(d)-----------------------------------------------------------
probability of observing an age of 80 or higher: 0.09720694972378524
```

- source code

```
print(f"(d)-----------------------------------------------------------")
x_variance = np.var(X)
x_SD = math.sqrt(x_variance)
x = np.arange(x_mean - 4 * x_SD - 1, x_mean + 4 * x_SD + 1, 0.001)
cdf = stats.norm(x_mean, x_SD).cdf(x)
plt.figure(figsize=(15, 10))
plt.title('cdf')
plt.xlabel('x')
plt.ylabel('Probability')
plt.xticks(np.arange(x_mean - 4 * x_SD, x_mean + 4 * x_SD + 0.001, x_SD),
           labels=['m-4∂', 'm-3∂', 'm-2∂', 'm-1∂', 'm', 'm+∂', 'm+2∂', 'm+3∂', 'm+4∂'])
plt.grid()
# plt.plot(x, stats.norm.pdf(x, loc=x_mean, scale=x_SD))
plt.fill_between(x[x <= 80], cdf[x <= 80], alpha=0.5)
plt.fill_between(x[x >= 80], cdf[x >= 80])
print(f"probability of observing an age of 80 or higher: {stats.norm.cdf(1 - (80 - x_mean) / x_SD)}")
```

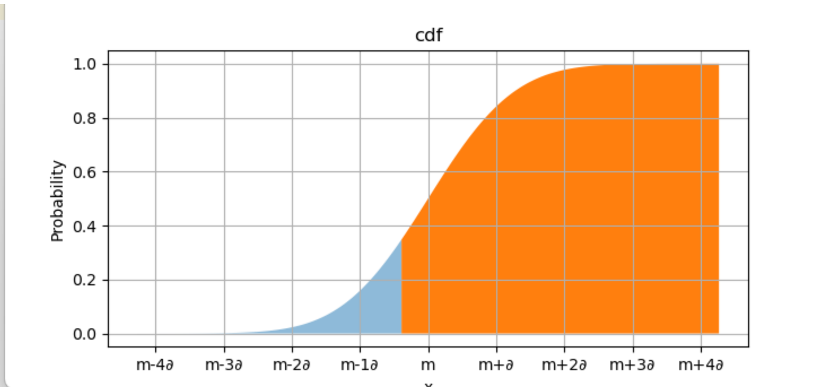
n()

main ×

```
C:\Users\iqeq1\anaconda3\python.exe "C:\Users\iqeq1\Downloads\code examples-20230314\main.py"
(d)-----------------------------------------------------------
probability of observing an age of 80 or higher: 0.09720694972378524
```

(e) What is the probability of observing an age of 70 or higher?

- Calculate by z-score method and graph




```
D),
'm+3∂', 'm+4∂'])
plt.fill_between(x[x <= 70], cdf[x <= 70], alpha=0.5)
plt.fill_between(x[x >= 70], cdf[x >= 70])
print(f"probability of observing an age of 70 or higher: {stats.norm.cdf(1 - (70 - x_mean) / x_SD)}")
```

```
C:\Users\iqeq1\anaconda3\python.exe "C:\Users\iqeq1\Downloads\code examples-20230314\main.py"
(e)------------------------------------------------------------
probability of observing an age of 70 or higher: 0.9176834481530523
```

- source code

```
print(f"(e)------------------------------------------------------------")
x_variance = np.var(X)
x_SD = math.sqrt(x_variance)
x = np.arange(x_mean - 4 * x_SD - 1, x_mean + 4 * x_SD + 1, 0.001)
cdf = stats.norm(x_mean, x_SD).cdf(x)
plt.figure(figsize=(15, 10))
plt.title('cdf')
plt.xlabel('x')
plt.ylabel('Probability')
plt.xticks(np.arange(x_mean - 4 * x_SD, x_mean + 4 * x_SD + 0.001, x_SD),
           labels=['m-4∂', 'm-3∂', 'm-2∂', 'm-1∂', 'm', 'm+∂', 'm+2∂', 'm+3∂', 'm+4∂'])
plt.grid()
# plt.plot(x, stats.norm.pdf(x, loc=x_mean, scale=x_SD))
plt.fill_between(x[x <= 70], cdf[x <= 70], alpha=0.5)
plt.fill_between(x[x >= 70], cdf[x >= 70])
print(f"probability of observing an age of 70 or higher: {stats.norm.cdf(1 - (70 - x_mean) / x_SD)}")
```

```
C:\Users\iqeq1\anaconda3\python.exe "C:\Users\iqeq1\Downloads\code examples-20230314\main.py"
(e)------------------------------------------------------------
probability of observing an age of 70 or higher: 0.9176834481530523
```

(f) Find the 2-dimensional mean vector and the covariance matrix for these two variables.

```
    X = [69, 74, 68, 70, 72, 67, 66, 70, 76, 68, 72, 79, 74, 67, 66, 71, 74, 75, 75, 76]
    Y = [153, 175, 155, 135, 172, 150, 115, 137, 200, 130, 140, 265, 185, 112, 140, 150, 165, 185, 210, 220]
    print(f"(f)--------------------------------------------------------")
    x_mean, y_mean = np.mean(X), np.mean(Y)
    x_variance, y_variance = np.var(X), np.var(Y)
    cov_hw = np.cov(X, Y, ddof=1)
    print(f"x_mean, y_mean : {x_mean}, {y_mean}") # x_mean, y_mean : 71.45, 164.7
    print(f"2-dimensional mean vector = [x_mean, y_mean] = : [{x_mean}, {y_mean}]") #  [x_mean, y_mean] = : [71.45, 164.7]
    print(f"covariance matrix =\n {cov_hw}") #  [[  14.57631579  128.87894737] [ 128.87894737 1441.27368421]]
if __name__ =='__main__':
    main()
```

```
C:\Users\iqeq1\anaconda3\python.exe "C:\Users\iqeq1\Downloads\code examples-20230314\main.py"
(f)-----------------------------------------------------------
x_mean, y_mean : 71.45, 164.7
2-dimensional mean vector = [x_mean, y_mean] = : [71.45, 164.7]
covariance matrix =
 [[  14.57631579  128.87894737]
 [ 128.87894737 1441.27368421]]
```

-covariance matrix

≒ [[14.58   128.88]

[128.88 1441.27]]

Task-2

Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:

age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

(a) Calculate the mean, median, and standard deviation of *age* and *%fat*.

(b) Draw the boxplots for *age* and *%fat*.

(c) Draw a *scatter plot* and a *q-q plot* based on these two variables.

Ans:

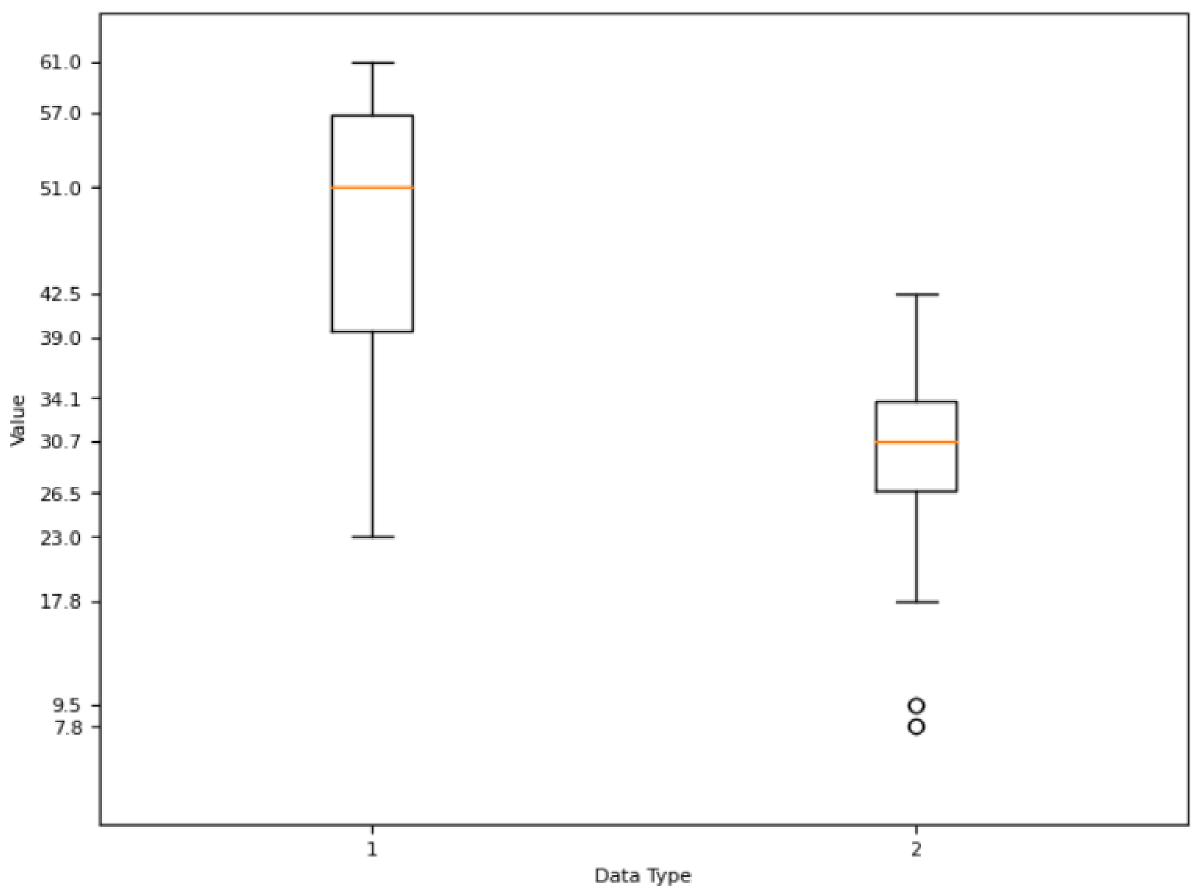
(a) age: mean=46.44, median=51, standard deviation=12.85

%fat: mean=28.78, median=30.7, standard deviation=8.99

```
In [2]: import matplotlib.pyplot as plt
        import math
        import numpy as np
        from scipy import stats

        age = [23,23,27,27,39,41,47,49,50,52,54,54,56,57,58,58,60,61]
        fat_ratio = [9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 27.2,
                     31.2, 34.6, 42.5, 28.8, 33.4, 30.2, 34.1, 32.9, 41.2, 35.7]
        print(f"(a)------------------------------------------------------------")
        age_mean,fat_mean =np.mean(age), np.mean(fat_ratio)
        age_median,fat_median  = np.median(age), np.median(fat_ratio)
        age_SD, fat_SD = math.sqrt(np.var(age)), math.sqrt(np.var(fat_ratio))

        print(f"age : mean={age_mean}, median={age_median}, standard deviation={age_SD} ") # mean=46.444, median=51.0, standard deviation=12.84
        print(f"%fat : mean={fat_mean}, median={fat_median}, standard deviation={fat_SD}") # mean=28.783, median=30.7, standard deviation=8.99

        (a)------------------------------------------------------------
        age : mean=46.44444444444444, median=51.0, standard deviation=12.846193652519204
        %fat : mean=28.783333333333335, median=30.7, standard deviation=8.993655170915401
```

(b) 1 : age 2 : %fat

```
print(f"(b)-----------------------------------------------------------")
plt.style.use('default')
plt.rcParams['figure.figsize'] = (8, 6)
plt.rcParams['font.size'] = 8

data_age = age
data_fat = fat_ratio

fig, ax = plt.subplots()

plt.xticks([1,2], labels=['age', 'fat'])
plt.yticks([7.8, 9.5, 17.8, 23, 26.5, 30.7,34.1, 39, 42.5, 51,57, 61])
ax.boxplot([data_age, data_fat])
ax.set_ylim(0.0, 65.0)
ax.set_xlabel('Data Type')
ax.set_ylabel('Value')

plt.show()
```

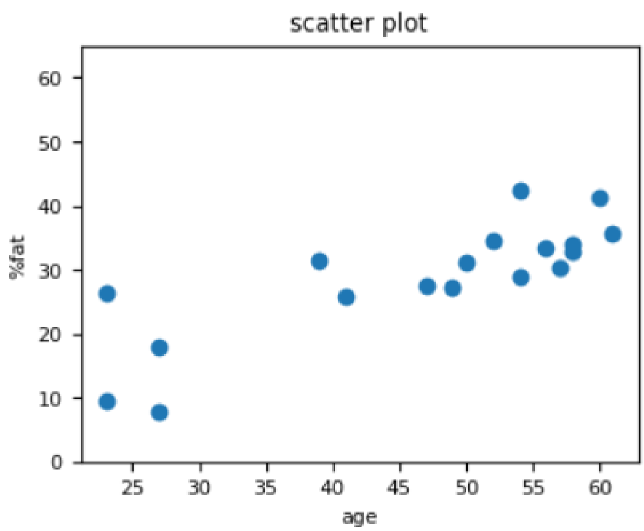


(c)

- scatter plot

```
print(f"(c)------------------------------------------------------------")
plt.style.use('default')
plt.rcParams['figure.figsize'] = (4, 3)
plt.rcParams['font.size'] = 8

data_age = age
data_fat = fat_ratio

fig, ax = plt.subplots()


ax.set_xlabel('age')
ax.set_ylabel('%fat')
plt.scatter(data_age, data_fat)
ax.set_ylim(0.0, 65.0)
ax.set_title('scatter plot')
plt.show()
```

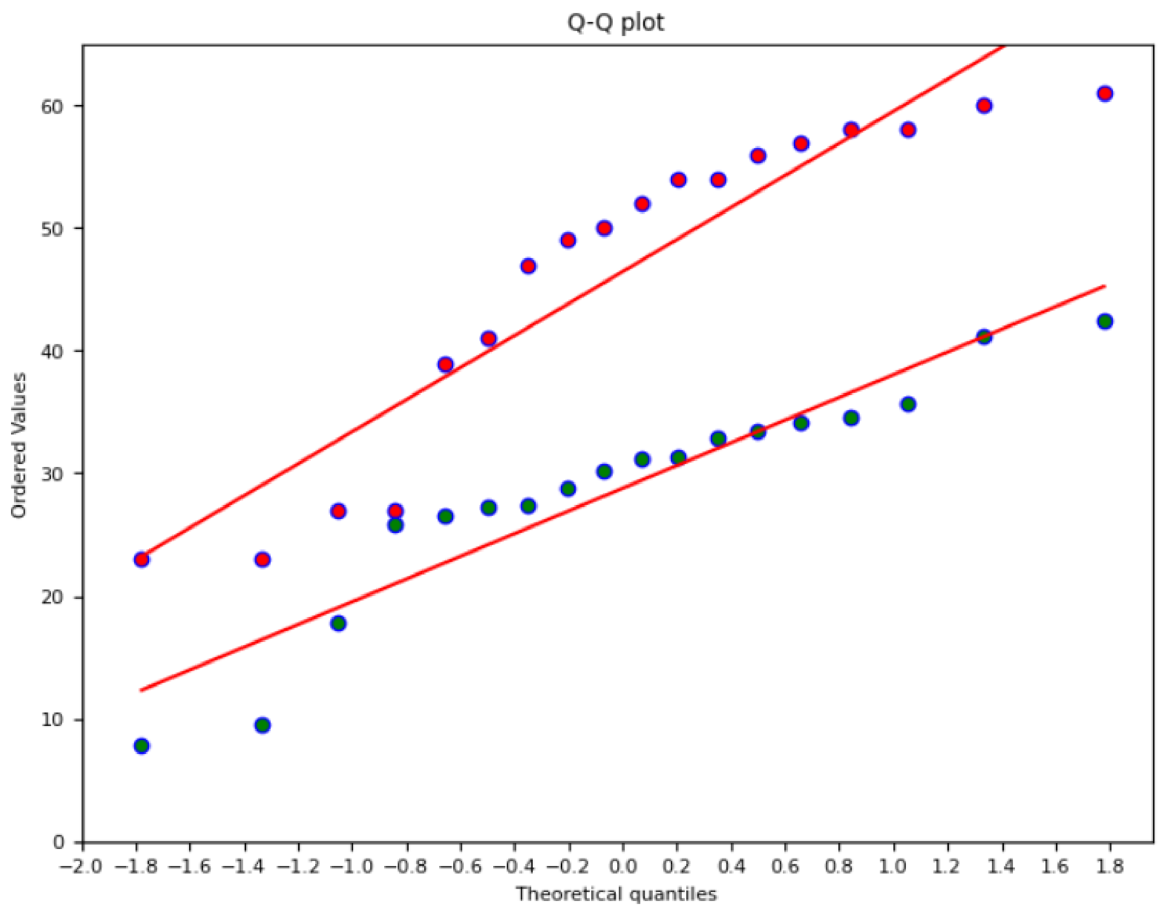
```
(c)------------------------------------------------------------
```



- q-q plot

```
print(f"(c)------------------------------------------------------------")
plt.style.use('default')
plt.rcParams['figure.figsize'] = (8, 6)
plt.rcParams['font.size'] = 8

data_age = age
data_fat = fat_ratio

fig, ax = plt.subplots()

stats.probplot(data_age, plot=plt)
stats.probplot(data_fat, plot=plt)
ax.get_lines()[0].set_markerfacecolor('r') # data_age
ax.get_lines()[2].set_markerfacecolor('g') # data_fat
ax.set_ylim(0.0, 65.0)
ax.set_title('Q-Q plot')
plt.xticks(np.arange(-2,2, step=0.2))
plt.show()
```

(c)--



Task-3

Suppose we have the following 2-D data set:

	A_1	A_2
x_1	1.5	1.7
x_2	2	1.9
x_3	1.6	1.8
x_4	1.2	1.5
x_5	1.5	1.0

(a) Consider the data as 2-D data points. Given a new data point, $\boldsymbol{x} = (1.4, 1.6)$ as a query, rank the database points based on similarity with the query using Euclidean distance, Manhattan distance, supremum distance, and cosine similarity.

(b) Normalize the data set to make the norm of each data point equal to 1. Use Euclidean distance on the transformed data to rank the data points.

Ans:

Write your Answer here

(a)

- Euclidean distance(ED(A,B))

$$\boldsymbol{ED(x1, x) = \sqrt{(1.5 - 1.4)^2 + (1.7 - 1.6)^2} = \sqrt{0.02}}$$

$$\boldsymbol{ED(x2, x) = \sqrt{(2 - 1.4)^2 + (1.9 - 1.6)^2} = \sqrt{0.45}}$$

$$\boldsymbol{ED(x3, x) = \sqrt{(1.6 - 1.4)^2 + (1.8 - 1.6)^2} = \sqrt{0.08}}$$

$$\boldsymbol{ED(x4, x) = \sqrt{(1.2 - 1.4)^2 + (1.5 - 1.6)^2} = \sqrt{0.05}}$$

$$\boldsymbol{ED(x5, x) = \sqrt{(1.5 - 1.4)^2 + (1.0 - 1.6)^2} = \sqrt{0.37}}$$

x1, x4, x3, x5, x2

$$\mathbf{MD(x1, x) = |1.5 - 1.4| + |1.7 - 1.6| = 0.2}$$

$$\mathbf{MD(x2, x) = |2 - 1.4| + |1.9 - 1.6| = 0.9}$$

$$\mathbf{MD(x3, x) = |1.6 - 1.4| + |1.8 - 1.6| = 0.4}$$

$$\mathbf{MD(x4, x) = |1.2 - 1.4| + |1.5 - 1.6| = 0.3}$$

$$\mathbf{MD(x5, x) = |1.5 - 1.4| + |1.0 - 1.6| = 0.7}$$

- Manhattan distance(MD(A,B))

x1, x4, x3, x5, x2

- supremum distance(SD(A,B))

$$\mathbf{SD(x1, x)} = \boldsymbol{max}(|\mathbf{1.5} - \mathbf{1.4}\,|, |\mathbf{1.7} - \mathbf{1.6}|) = \mathbf{0.1}$$

$$\mathbf{SD(x2, x)} = \boldsymbol{max}(|\mathbf{2} - \mathbf{1.4}\,|, |\mathbf{1.9} - \mathbf{1.6}|) = \mathbf{0.6}$$

$$\mathbf{SD(x3, x)} = \boldsymbol{max}(|\mathbf{1.6} - \mathbf{1.4}\,|, |\mathbf{1.8} - \mathbf{1.6}|) = \mathbf{0.2}$$

$$\mathbf{SD(x4, x)} = \boldsymbol{max}(|\mathbf{1.2} - \mathbf{1.4}\,|, |\mathbf{1.5} - \mathbf{1.6}|) = \mathbf{0.2}$$

$$\mathbf{SD(x5, x)} = \boldsymbol{max}(|\mathbf{1.5} - \mathbf{1.4}\,|, |\mathbf{1.0} - \mathbf{1.6}|) = \mathbf{0.6}$$

x1, x3=x4, 2=x5

- cosine similarity (CS(A, B))

[21]:

```
import numpy as np
from numpy import dot
from numpy.linalg import norm

def cos_sim(A, B):
  return dot(A, B)/(norm(A)*norm(B))

x1 = np.array([1.5,1.7])
x2 = np.array([2.0,1.9])
x3 = np.array([1.6,1.8])
x4 = np.array([1.2,1.5])
x5 = np.array([1.5,1.0])
x = np.array([1.4,1.6])



print('x1 :',cos_sim(x1, x))
print('x2 :',cos_sim(x2, x))
print('x3 :',cos_sim(x3, x))
print('x4 :',cos_sim(x4, x))
print('x5 :',cos_sim(x5, x))
```

```
x1 : 0.999991391443956
x2 : 0.9957522612528874
x3 : 0.9999694838187877
x4 : 0.9990282349375618
x5 : 0.9653633930282662
```

x1,x3,x4,x2,x5

(b)

- Normalizing the data

  x : (0.6585, 0.7526)

  x1: (0.6616, 0.7498)

  x2: (0.7250, 0.6887)

  x3: (0.6644, 0.7474)

  x4: (0.6247, 0.7809)

  x5: (0.8321, 0.5547)

- New eucildean distance

  x1: (0.0041)

  x2: (0.0922)

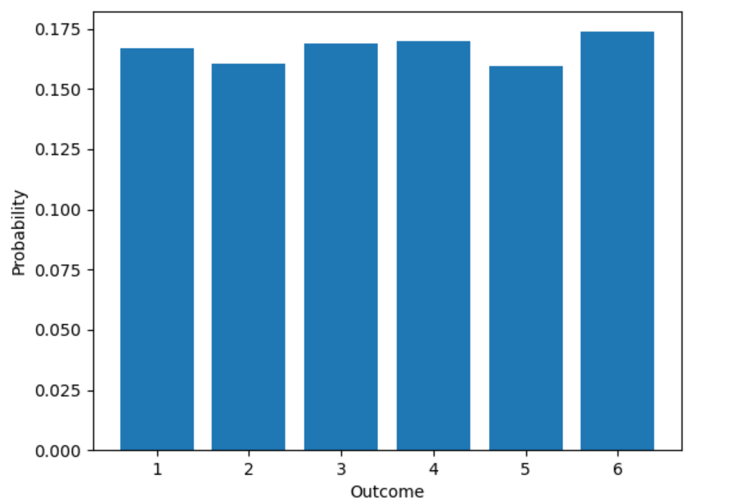
  x3: (0.0078)

  x4: (0.0441)

  x5: (0.2632)

Task-4: Run the Python code and record the results.

=prob01()

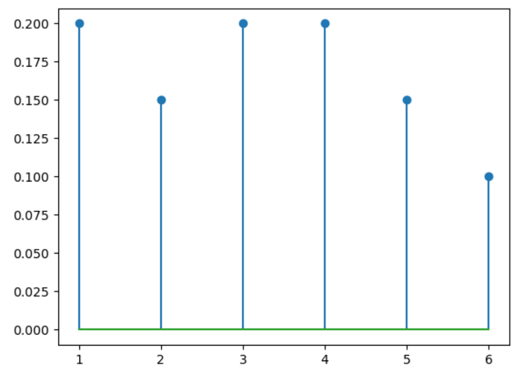


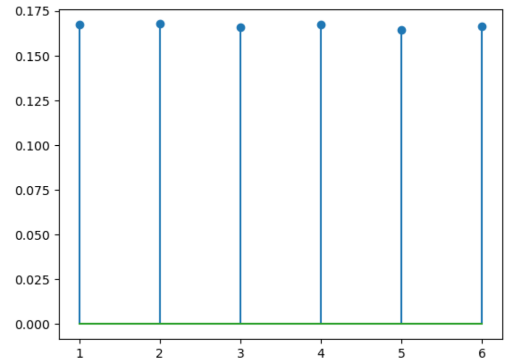
1.py Task-1.csv iris.csv main.py

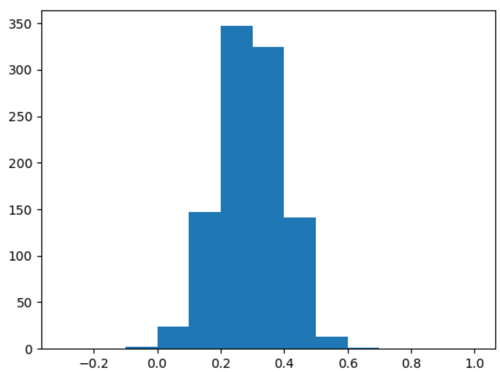
```
from lab01 import *
from distributions import *

def main():
    #run_distbutions()
    prob01()
    #prob02()
    #prob03()
    #dataA01()
    #dataA02()
    #dataA03()

if __name__ =='__main__':
    main()
```

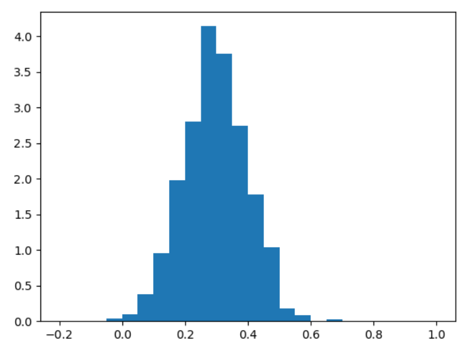
```
C:\Users\iqeq1\anaconda3\python.exe "C:\Users\iqeq1\Downloads\code examples-20230314\main.py"
[0.1671 0.1606 0.1691 0.17   0.1595 0.1737]
```

=prob02()

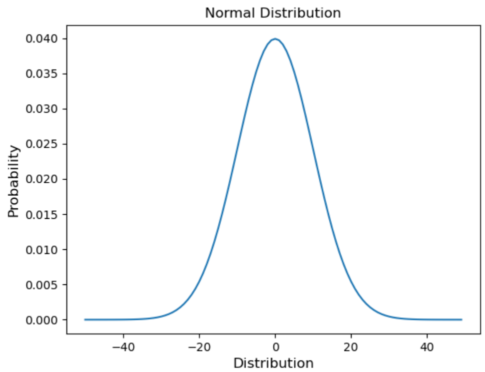

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
1 2 3 4 5 6

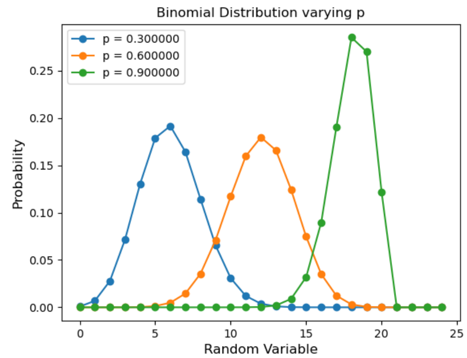

0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
1 2 3 4 5 6

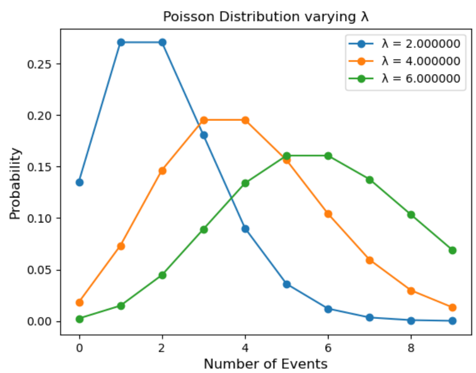

350
300
250
200
150
100
50
0
−0.2 0.0 0.2 0.4 0.6 0.8 1.0

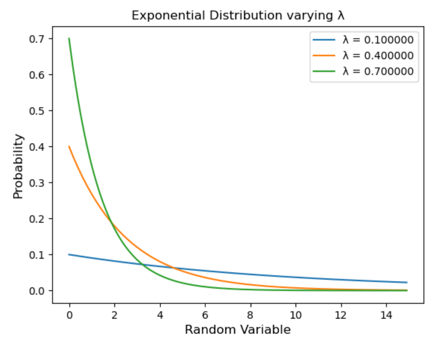

4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.2 0.0 0.2 0.4 0.6 0.8 1.0

=prob03()

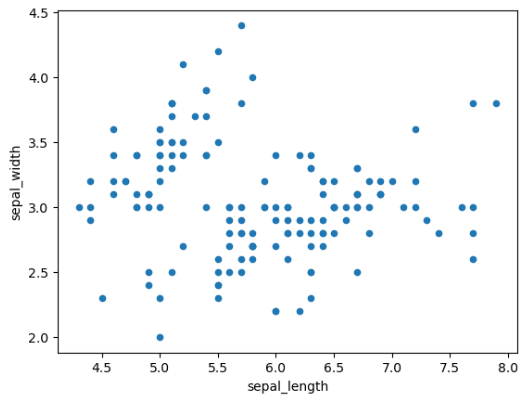

Normal Distribution
Probability
Distribution

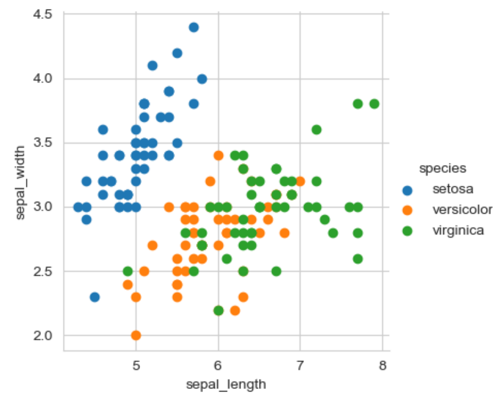

Binomial Distribution varying p
p = 0.300000
p = 0.600000
p = 0.900000
Probability
Random Variable


Poisson Distribution varying λ
λ = 2.000000
λ = 4.000000
λ = 6.000000
Probability
Number of Events


Exponential Distribution varying λ
λ = 0.100000
λ = 0.400000
λ = 0.700000
Probability
Random Variable

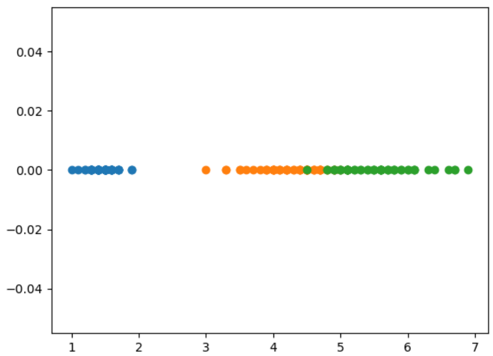
=dataA01()

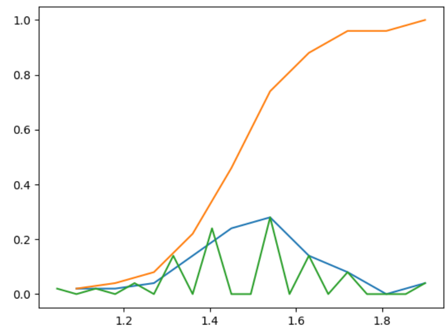


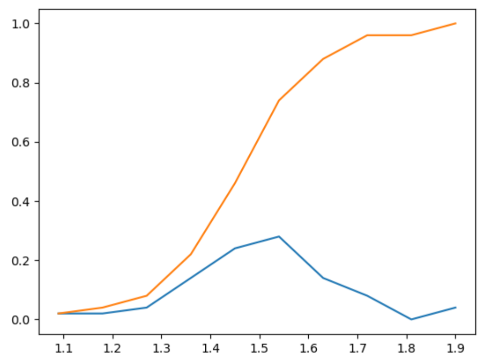


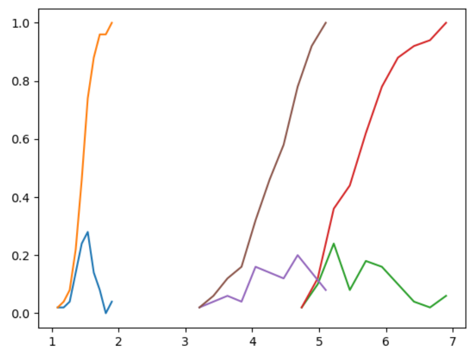

```
(150, 5)
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
        sepal_length  sepal_width  petal_length  petal_width species
count     150.000000   150.000000    150.000000   150.000000     150
unique           NaN          NaN           NaN          NaN       3
top              NaN          NaN           NaN          NaN  setosa
freq             NaN          NaN           NaN          NaN      50
mean        5.843333     3.054000      3.758667     1.198667     NaN
std         0.828066     0.433594      1.764420     0.763161     NaN
min         4.300000     2.000000      1.000000     0.100000     NaN
25%         5.100000     2.800000      1.600000     0.300000     NaN
50%         5.800000     3.000000      4.350000     1.300000     NaN
75%         6.400000     3.300000      5.100000     1.800000     NaN
max         7.900000     4.400000      6.900000     2.500000     NaN
setosa        50
versicolor    50
virginica     50
Name: species, dtype: int64
```

=dataA02()










```
[0.02 0.02 0.04 0.14 0.24 0.28 0.14 0.08 0.   0.04]
[1.   1.09 1.18 1.27 1.36 1.45 1.54 1.63 1.72 1.81 1.9 ]
[0.02 0.02 0.04 0.14 0.24 0.28 0.14 0.08 0.   0.04]
[1.   1.09 1.18 1.27 1.36 1.45 1.54 1.63 1.72 1.81 1.9 ]
[0.02 0.02 0.04 0.14 0.24 0.28 0.14 0.08 0.   0.04]
[1.   1.09 1.18 1.27 1.36 1.45 1.54 1.63 1.72 1.81 1.9 ]
[0.02 0.1  0.24 0.08 0.18 0.16 0.1  0.04 0.02 0.06]
[4.5  4.74 4.98 5.22 5.46 5.7  5.94 6.18 6.42 6.66 6.9 ]
[0.02 0.04 0.06 0.04 0.16 0.14 0.12 0.2  0.14 0.08]
[3.   3.21 3.42 3.63 3.84 4.05 4.26 4.47 4.68 4.89 5.1 ]
```

=dataA03()

```
Means:
1.464
2.4156862745098038
5.552
4.26

Std-dev:
0.17176728442867115
0.5463478745268441
0.4651881339845204

Medians:
1.5
1.5
5.55
4.35

Quantiles:
[1.    1.4   1.5   1.575]
[4.5   5.1   5.55  5.875]
[3.   4.   4.35 4.6 ]

90th Percentiles:
1.7
6.31
4.8

Median Absolute Deviation
0.14826022185056031
0.6671709983275211
0.5189107764769602
```

Comments

Write any comments about the task. For example, what you have learnt from the task, it was helpful in understanding concepts etc.

Thank you for the fun assignment.

Thanks to that, I was able to review linear algebra and pyplot.