

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по практической работе №1**  
**по дисциплине «Машинное обучение»**

Студент гр. 1310

Комаров Д. Е.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2025

## **Задание 1**

### **Постановка задачи**

Предположим, X и Y две случайные переменные, отражающие возраст и вес соответственно. Рассмотрим случайную выборку из 20 наблюдений:

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).

Необходимо:

- A. найти среднее, медиану и моду величины X;
- B. найти дисперсию Y;
- C. построить график нормального распределения для X;
- D. найти вероятность того, что возраст больше 80;
- E. найти двумерное мат. ожидания и ковариационную матрицу для этих двух величин;
- F. определять корреляцию между X и Y;
- G. построить диаграмму рассеяния, отображающая зависимость между возрастом и весом.

### **Код программы**

```
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as scst

X=np.array([69, 74, 68, 70, 72, 67, 66, 70, 76, 68, 72, 79, 74, 67, 66, 71, 74, 75, 75, 76])
Y=np.array([153, 175, 155, 135, 172, 150, 115, 137, 200, 130, 140, 265, 185, 112, 140, 150, 165, 185, 210, 220])

#A
print("Mean X: %.2f" % np.mean(X))
print("Median X: %.2f" % np.median(X))
print("Mode X: %.2f" % scst.mode(X).mode)

#B
print("Variance Y: %.2f" % np.var(Y, ddof=1))

#C
distX=scst.norm(loc=np.mean(X), scale=np.std(X, ddof=1))
minX=distX.mean()-3*distX.std()
maxX=distX.mean()+3*distX.std()
step=0.1
pointsx=[]
```

```

pointsf=[]
for x in (minX+i*step for i in range(int((maxX-minX)/step))): 
    pointsx.append(x)
    pointsf.append(distX.pdf(x))
plt.plot(pointsx,pointsf)
plt.grid()
plt.xlabel("x")
plt.ylabel("f(x) ")
plt.show()

#D
print("The probability that the age is greater than 80: %.2f" % (1-distX.cdf(80)))

#E
print("Covariance matrix:\n", np.cov(X,Y))

#F
print("Correlation: %.2f" % np.corrcoef(X,Y)[0][1])

#G
plt.scatter(X,Y)
plt.xlabel("X (возраст) ")
plt.ylabel("Y (вес) ")
plt.grid()
plt.show()

```

## Результат выполнения

A. Среднее X: 71.45; медиана X: 71.5; мода X: 74.

B. Дисперсия Y: 1441.27

C. График нормального распределения для X представлен на рисунке 1.

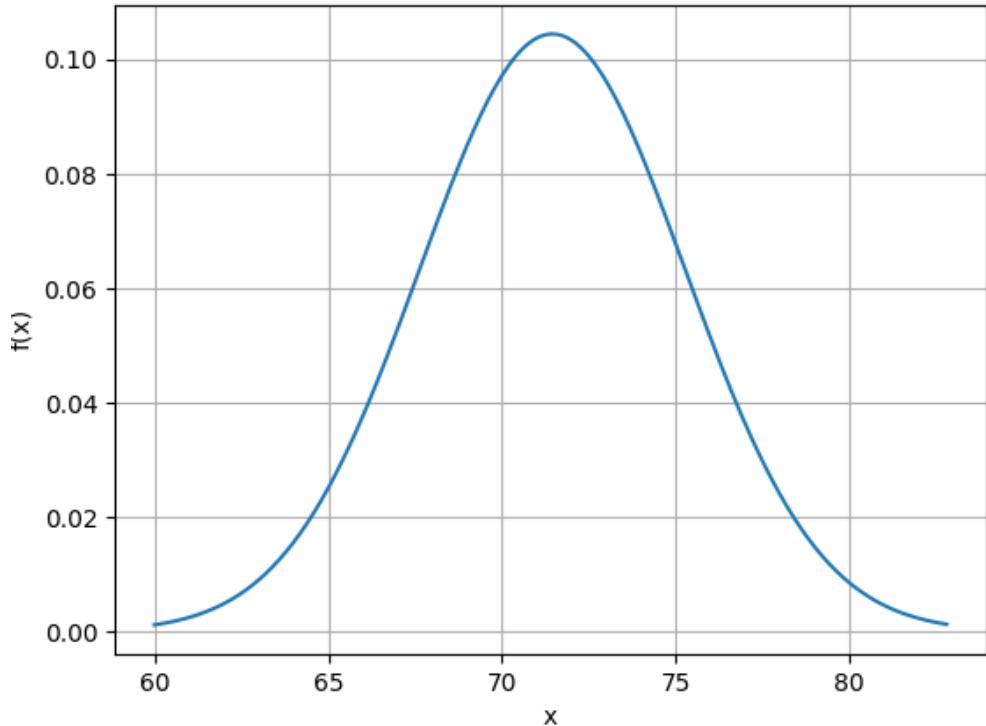


Рисунок 1 – График нормального распределения для X

D. Вероятность того, что возраст больше 80: 0.01

E. Ковариационная матрица для величин X и Y

$$\begin{pmatrix} 14.58 & 128.88 \\ 128.89 & 1441.27 \end{pmatrix}.$$

F. Корреляция между X и Y: 0.89

G. Диаграмма рассеяния, отображающая зависимость между возрастом и весом представлена на рисунке 2.

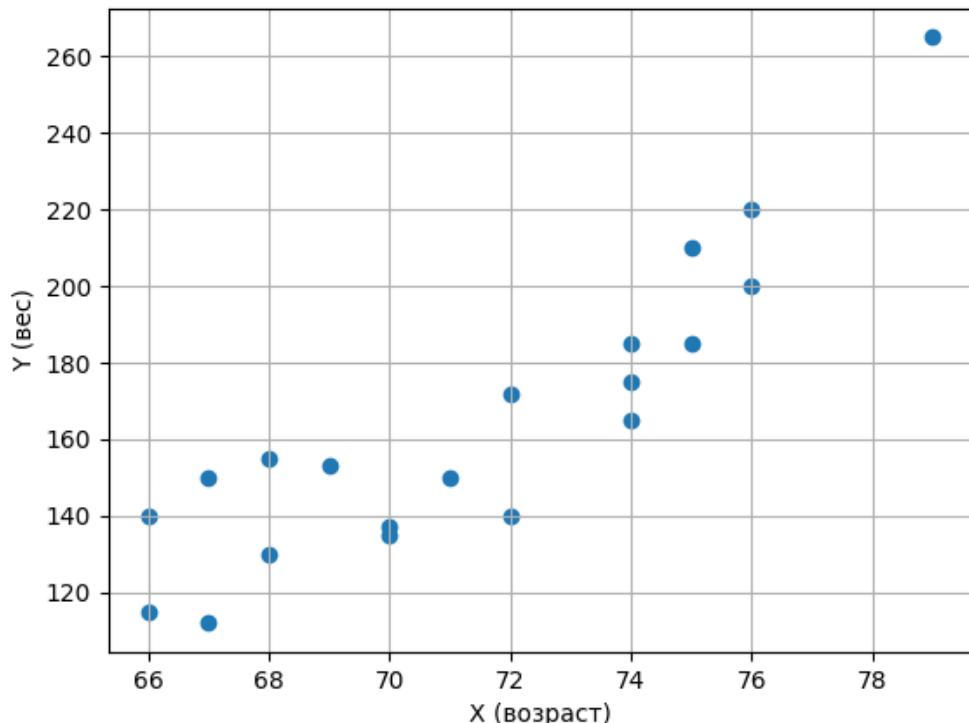


Рисунок 2 – Диаграмма рассеяния, отображающая зависимость между возрастом и весом

## Задание 2

### Постановка задачи

Для набора данных, представленного в таблице 1 рассчитайте ковариационную матрицу и обобщенную дисперсию.

Таблица 1 – Набор данных

	X1	X2	X3
a	17	17	12
b	11	9	13
c	11	8	19

### Код программы

```
import numpy as np

X=np.array([
    [17,17,12],
    [11,9,13],
    [11,8,19]
]).transpose()
covmat=np.cov(X)
print("Covariance matrix:\n",covmat)
print("Generalized variance: ", np.linalg.det(covmat))
```

### Результат выполнения

Ковариационная матрица

$$\begin{pmatrix} 12 & 17 & -8 \\ 17 & 24.33 & -12.83 \\ -8 & -12.83 & 14.33 \end{pmatrix}.$$

Обобщенная дисперсия равна 0.

## Задание 3

### Постановка задачи

Даны два одномерных нормальных распределения  $Na$  и  $Nb$  с математическими ожиданиями 4, 8 и СКО 1, 2 соответственно.

1. Для каждого из значения  $\{5,6,7\}$  определите какое из распределений сгенерировало значение с большей вероятностью.
2. Найди значение, которой могло быть сгенерировано обеими распределениями с равной вероятностью.

### Код программы

```
import scipy.stats as scst
import matplotlib.pyplot as plt

Nadist=scst.norm(loc=4,scale=1)
Nbdist=scst.norm(loc=8,scale=2)
#A
points=[5,6,7]
for p in points:
    print(p,"most likely generated by "+"("+"Na" if Nadist.pdf(p)> Nbdist.pdf(p)
else "Nb")+" distribution")
#B
minX=1
maxX=14
step=0.1
pointsNa=[]
pointsNb=[]
pointsx=[]
for x in (minX+i*step for i in range(int((maxX-minX) / step))): 
    pointsx.append(x)
    pointsNa.append(Nadist.pdf(x))
    pointsNb.append(Nbdist.pdf(x))
plt.plot(pointsx,pointsNa)
plt.plot(pointsx,pointsNb)
plt.show()
xst=5
xed=6
step=0.01
delta=1
resx=0
for x in (i*step+xst for i in range(int((xed-xst) / step))): 
    d=abs(Nadist.pdf(x)-Nbdist.pdf(x))
    if d<delta:
        delta,resx=d,x
print(resx,"can be generated by both distributions with equal probability")
```

### Результат выполнения

1. Значение 5 вероятнее принадлежит к выборке  $Na$ , значения 6 и 7 вероятнее принадлежат к выборке  $Nb$ .
2. Значение 5.66 может быть сгенерировано обеими выборками с равной вероятностью.