

Задание 1

Входные данные:

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

Среднее значение, медиана и мода X. Листинг программы:

```
import numpy as np

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])

mean_x = x.mean()
median_x = np.median(x)

(_, idx, counts) = np.unique(x, return_index=True, return_counts=True)
index = idx[np.argmax(counts)]
mode_x = x[index]

print(f'Среднее значение: {mean_x}')
print(f'Медиана: {median_x}')
print(f'Мода: {mode_x}')
```

Ответ:

Среднее значение: 71.45

Медиана: 71.5

Мода: 74

B

Дисперсия Y:

```
print(f'Дисперсия Y: {y.var()}')
```

Ответ:

Дисперсия Y: 1369.2099999999998

С

Распределение X отличается от нормального, что показано на гистограмме на рис. 1. Необходимо построить случайную величину с нормальным распределением, сохранив математическое ожидание и дисперсию.

```
plt.hist(x, bins=8)
```

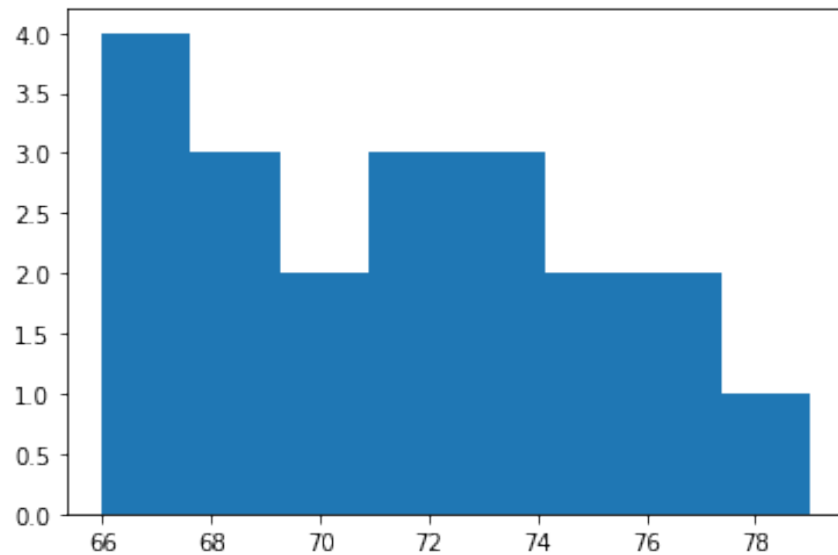


Рис. 1. Гистограмма случайной величины X .

Построение нормального распределения:

```
x_axis = np.linspace(x.min(), x.max())  
y_axis = stats.norm.pdf(x_axis, x.mean(), x.std())  
  
plt.plot(x_ax, y_ax)  
plt.grid()
```

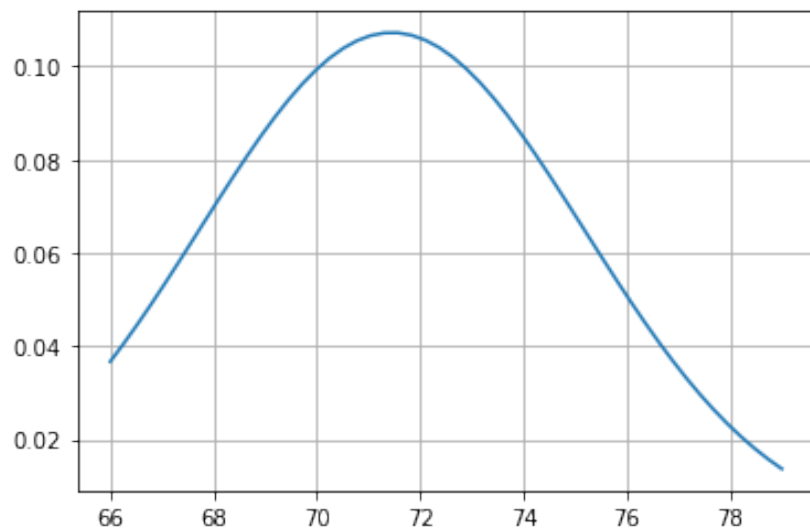


Рис. 2. Нормальное распределение для X.

D

Вероятность того, что возраст выше 80:

```
print(f'Возраст меньше 80 с вероятностью: {np.count_nonzero(x > 80) /
len(x)}')
```

Ответ:

Возраст меньше 80 с вероятностью: 0.0

E

Нахождение двумерного матожидания и матрицы ковариации для X и Y:

```
print(f'Двумерное матожидание: {np.mean([x, y], axis=1)}')
print(f'Матрица ковариации: {np.cov([x, y])}')
```

Ответ:

Двумерное матожидание: [71.45 164.7]

Матрица ковариации: [[14.57631579 128.87894737]
[128.87894737 1441.27368421]]

F

Ковариация между X и Y:

```
print(f'Коэффициент ковариации X и Y: {np.corrcoef(x, y)[0, 1]}')
```

Ответ:

Коэффициент ковариации X и Y: 0.8891701351748048

G

Диаграмма рассеяния, показывающая зависимость между возрастом и весом, показана на рис. 3.

```
plt.scatter(x, y)
plt.grid()
plt.xlabel('Возраст')
plt.ylabel('Вес')
```

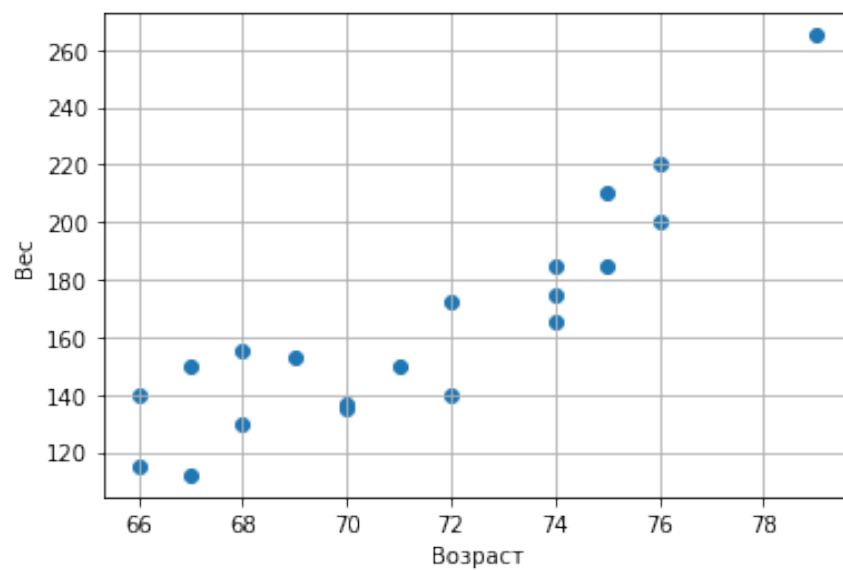


Рис. 3. Диаграмма рассеяния для случайных величин X и Y.

Задание 2

Набор данных:

	X ₁	X ₂	X ₃
a	17	17	12
b	11	9	13
c	11	8	19

Расчёт ковариационной матрицы и дисперсии:

```
data = np.array([
    [17, 17, 12],
```

```

        [11,  9, 13],
        [11,  8, 19]
    ])

cov_matrix = np.cov(data.T)
display(tabulate(cov_matrix,
                 headers=['X_1', 'X_2', 'X_3'],
                 showindex=['X_1', 'X_2', 'X_3'],
                 tablefmt='html'))

print(f'Обобщённая дисперсия: {np.linalg.det(cov_matrix)}')
```

Ответ:

	X ₁	X ₂	X ₃
X ₁	12	17	-8
X ₂	17	23.33	-12.83
X ₃	-8	-12.83	14.33

Обобщённая дисперсия: 9.577387902356475e-14

Задание 3

Для двух нормальных распределений с математическими ожиданиями 4 и 8 и СКО 1 и 2 соответственно найти, какое сгенерировало заданные значения с большей вероятностью. Поскольку для реального нормального распределения вероятность выпадения конкретного значения равна нулю, для расчёта применим дискретизацию с шагом 1.

```

values = [5, 6, 7]

N = 10000
a_mean, b_mean = 4, 8
a_std, b_std = 1, 2

Na = np.random.normal(a_mean, a_std, N)
Nb = np.random.normal(b_mean, b_std, N)

probs = [[np.count_nonzero(np.abs(Na - x) < 1) / N,
          np.count_nonzero(np.abs(Nb - x) < 1) / N]
          for x in values]

display(tabulate(probs,
                 headers=['N_a', 'N_b'],
                 showindex=values,
```

```
tablefmt='html'))
```

Ответ:

	N_a	N_b
5	0.478	0.136
6	0.168	0.242
7	0.026	0.341

Для поиска значения, которое распределения сгенерируют с одинаковой вероятностью, приравняем функции распределения (график приведён на рис. 4).

```
x = np.linspace(0, 15, N)
pdf_a = stats.norm.pdf(x, 4, 1)
pdf_b = stats.norm.pdf(x, 8, 2)
plt.plot(x, pdf_a)
plt.plot(x, pdf_b)
plt.grid()
plt.legend(['$N_a$', '$N_b$'])

for i in range(N):
    if np.abs(pdf_a[i] - pdf_b[i]) < 0.001:
        print(f'Значение с равными вероятностями для обоих распределений:
{i / N * 15}')
        plt.plot(x[i], stats.norm.pdf(x[i], 4, 1), 'go')
```

Ответ:

Значение с равными вероятностями для обоих распределений: 5.6595

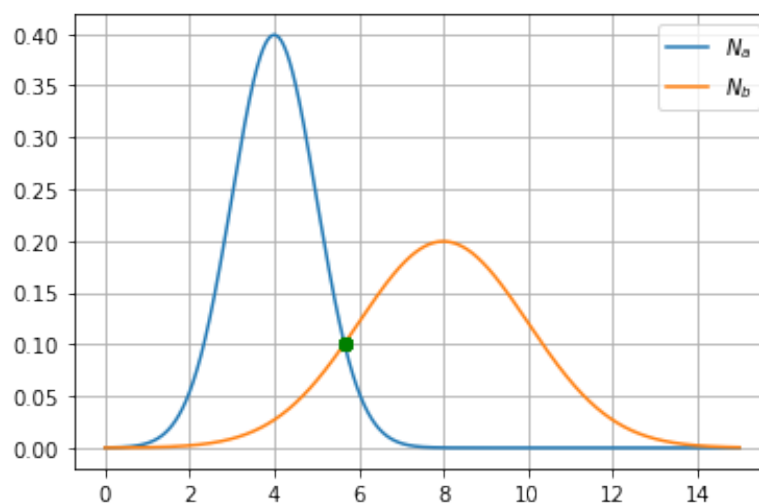


Рис. 4. Значение, генерируемое распределениями с равной вероятностью.