

Задание 1

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

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

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

1. Поиск среднего, медианы и моды величины X .

```
x_mean = X.mean()
x_median = np.median(X)
x_mode = stats.mode(X)[0]
print('Среднее x: ', x_mean, '\nМедиана x: ', x_median, '\nМода x: ', x_mode, '\n')
```

```
Среднее x:  71.45
Медиана x:  71.5
Мода x:  [74]
```

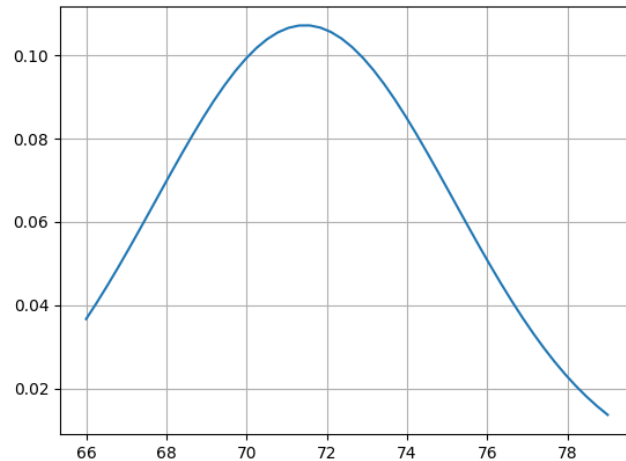
2. Найти дисперсию Y

```
y_var = np.var(Y)
print('Дисперсия Y', y_var, '\n')
```

```
Дисперсия Y 1369.2099999999998
```

3. Построить график нормального распределения для X

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



4. Найти вероятность того, что возраст больше 80

```
count80 = np.count_nonzero(X > 80)
print('Вероятность того, что возраст больше 80: ', count80, '%\n')
```

Вероятность того, что возраст больше 80: 0 %

5. Найти двумерное математическое ожидание и ковариационную матрицу для этих двух величин.

```
xy_mean = np.mean([X, Y], axis=1)
xy_cov = np.cov([X, Y])

print('Двумерное мат. ожидание: ', xy_mean)
print('Ковариационная матрица: \n', xy_cov, '\n')
```

Двумерное мат. ожидание: [71.45 164.7]

Ковариационная матрица:

```
[[ 14.57631579 128.87894737]
 [128.87894737 1441.27368421]]
```

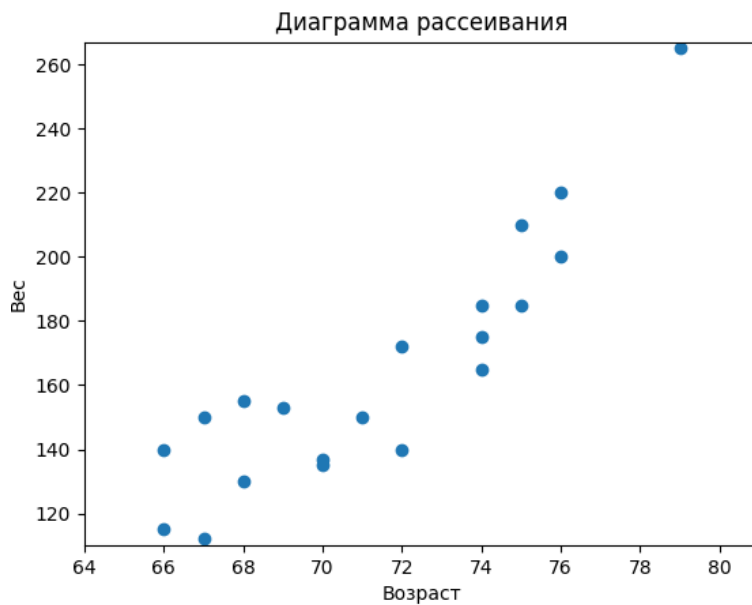
6. Определить корреляцию между X и Y

```
xy_corr = np.corrcoef(X, Y)[0, 1]
print('Коэффициент корреляции Пирсона', xy_corr, '\n')
```

Коэффициент корреляции Пирсона 0.8891701351748048

7. Построить диаграмму рассеяния, отображающую зависимость между возрастом и весом

```
plt.scatter(X, Y)
plt.title('Диаграмма рассеивания')
plt.xlabel('Возраст')
plt.ylabel('Вес')
plt.axis([X.min() - 2, X.max() + 2, Y.min() - 2, Y.max() + 2])
plt.show()
```



Задание 2

Рассчитать ковариационную матрицу и обобщенную дисперсию для следующего набора данных:

	X_1	X_2	X_3
a	17	17	12
b	11	9	13
c	11	8	19

```
matrix = np.array([
    [17, 17, 12],
    [11, 9, 13],
    [11, 8, 19]
])

# Ковариационная матрица
cov_matrix = np.cov(matrix.T)
print(cov_matrix)

# Обобщенная дисперсия – определитель км
gen_variance = np.linalg.det(cov_matrix)
print('Обобщенная дисперсия: ', gen_variance, '\n')

[[ 12.          17.          -8.          ]
 [ 17.          24.33333333 -12.83333333]
 [ -8.          -12.83333333  14.33333333]]
Обобщенная дисперсия:  0.0
```

Задание 3

Даны два одномерных нормальных распределения Na и Nb с мат. 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.

```
m_Na, m_Nb = 4, 8
std_Na, std_Nb = 1, 2
array = [5, 6, 7]
```

```
N = 100
```

```
Na = np.random.normal(m_Na, std_Na, N)
Nb = np.random.normal(m_Nb, std_Nb, N)
```

```
prob = [[np.count_nonzero(np.abs(Na - x) < 1) / N, np.count_nonzero(np.abs(Nb - x) < 1) / N]
         for x in array]
print('Вероятность генерации', prob)
```

Вероятность генерации [[0.5, 0.14], [0.14, 0.29], [0.03, 0.35]]

Следовательно, для 5 вероятность выше у Na, для 6 у Nb, для 7 у Nb

2. Найди значение, которой могло быть сгенерировано обеими распределениями с равной вероятностью

```
# Найти значение, которой могло быть сгенерировано обеими распределениями с равной вероятностью
res = root(lambda x: stats.norm.pdf(x, m_Nb, std_Nb) - stats.norm.pdf(x, m_Na, std_Na), 5)
print('Значение с равной вероятностью', res)
```

```
Значение с равной вероятностью      fjac: array([[ -1.]])
      fun: array([4.16333634e-17])
message: 'The solution converged.'
      nfev: 8
      qtf: array([4.49894288e-12])
      r: array([-0.22584348])
      status: 1
success: True
      x: array([5.65990966])
```

С явной вероятностью сгенерируется значение 5.65990966