

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

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

Студент гр. 6307

\_\_\_\_\_

Золотухин М. А.

Преподаватель

\_\_\_\_\_

Жангиров Т. Р.

Санкт-Петербург

2020

## Задание 1

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

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

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

### А

Найти среднее, медиану и моду величины  $X$ .

Решение:

```
import numpy
from scipy import stats

x = numpy.array([69, 74, 68, 70, 72, 67, 66, 70, 76, 68, 72, 79, 74, 67, 66, 71, 74, 75, 75, 76])

mean_x = x.mean()
median_x = numpy.median(x)
mode_x = stats.mode(x).mode[0]
```

Ответ:

```
X mean: 71.45
X median: 71.5
X mode: 74
```

### В

Найти дисперсию  $Y$ .

Решение:

```
y = numpy.array([153, 175, 155, 135, 172, 150, 115, 137, 200, 130, 140, 265, 185, 112, 140, 150, 165, 185, 210, 220])

variance_y = y.var()
```

Ответ:

```
Y variance: 1369.2099999999998
```

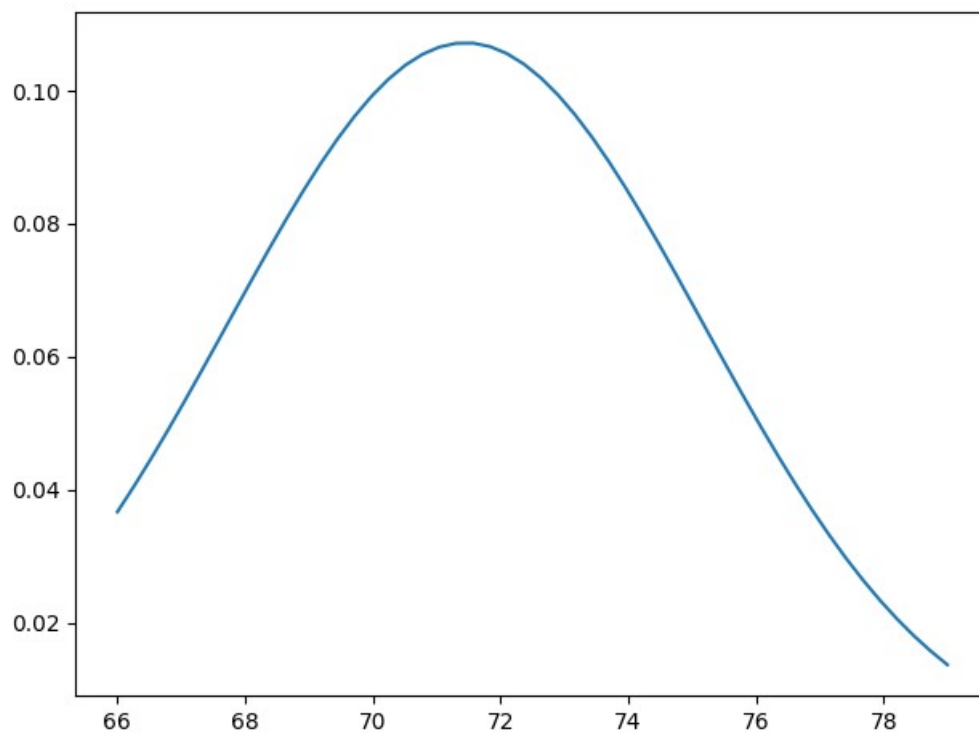
## C

Построить график нормального распределения для X

Решение:

```
from matplotlib import pyplot
std_x = x.std()
figure, axes = pyplot.subplots()
plot_x = numpy.linspace(x.min(), x.max())
plot_y = stats.norm.pdf(plot_x, mean_x, std_x)
pyplot.plot(plot_x, plot_y)
pyplot.show()
```

Ответ:



## D

Найти вероятность того, что возраст больше 80

Решение:

```
from scipy import integrate
def gauss_func_x(_x):
```

```
return stats.norm.pdf(_x, mean_x, std_x)

probability_that_x_gt_80 = integrate.quad(gauss_func_x, 80, mean_x + std_x * 3)
[0]
```

Ответ:

```
Probability that x > 80: 0.009441479887741383
```

## E

Найти двумерное мат. ожидания и ковариационную матрицу для этих двух величин

Решение:

```
mean_x = x.mean()
mean_y = y.mean()

cov_matrix = numpy.cov(x, y)
```

Ответ:

```
X mean: 71.45
Y mean: 164.7
Cov matrix of X and Y:
[[ 14.57631579 128.87894737]
 [128.87894737 1441.27368421]]
```

## F

Определять корреляцию между X и Y

Решение:

```
corr_x_y = numpy.corrcoef(x, y)[0][1]
```

Ответ:

```
X & Y correlation: 0.8891701351748048
```

## G

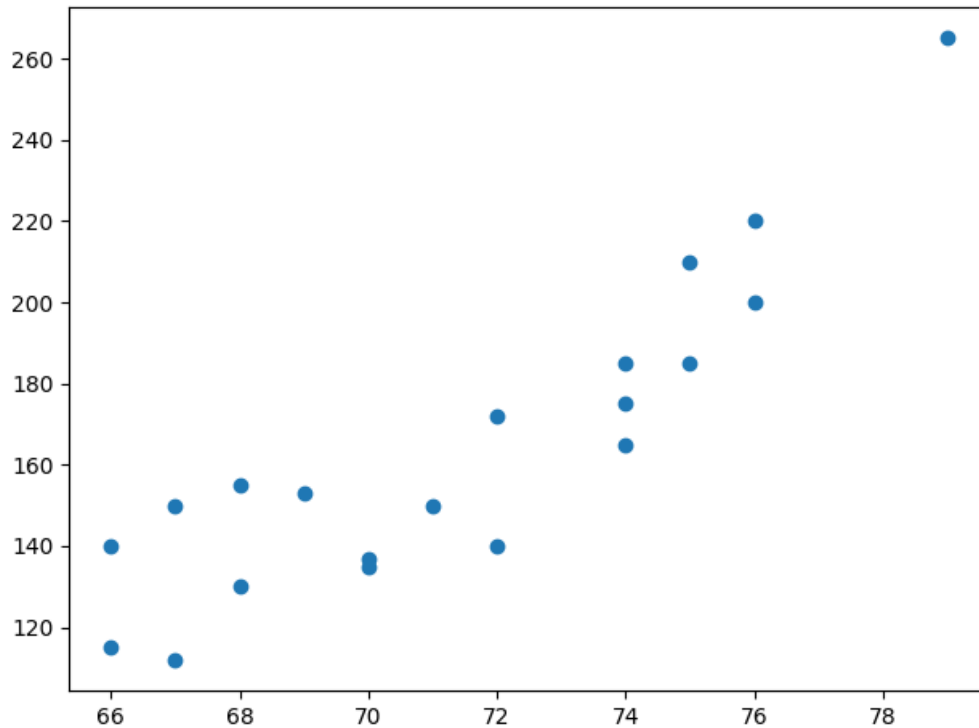
Построить диаграмму рассеяния, отображающая зависимость между возрастом и весом

Решение:

```
figure, axes = pyplot.subplots()
```

```
pyplot.scatter(x, y)
pyplot.show()
```

Ответ:



## Задание 2

Для следующего набора данных

X1 X2 X3

a 17 17 12

b 11 9 13

c 11 8 19

Рассчитайте ковариационную матрицу и обобщенную дисперсию

Решение:

```
def exercise_2():
    data = numpy.array([[17, 17, 12],
                        [11, 9, 13],
                        [11, 8, 19]])
```

```
cov_mat = numpy.cov(data)

print(f'Cov matrix: {cov_mat}')

general_variance = numpy.linalg.det(cov_mat)

print(f'General variance: {general_variance}')
```

Ответ:

```
Cov matrix: [[ 8.33333333 -5.         -15.83333333]
             [-5.         4.         11.         ]
             [-15.83333333 11.         32.33333333]]
General variance: 2.2204460492503156e-14
```

## Задание 3

Даны два одномерных нормальных распределения  $N_a$  и  $N_b$  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.
2. Найди значение, которой могло быть сгенерировано обеими распределениями с равной вероятностью

Решение:

```
def exercise_3():
    n_a_mean = 4
    n_a_std = 1

    n_b_mean = 8
    n_b_std = 4

    sample = [5, 6, 7]

    # 1. For each in sample: in which norm distribution is probability bigger?
    probability_sample_n_a = stats.norm.pdf(sample, n_a_mean, n_a_std)
    probability_sample_n_b = stats.norm.pdf(sample, n_b_mean, n_b_std)

    for index in range(0, len(sample)):
        print(f'Value: {sample[index]}')
        if probability_sample_n_a[index] > probability_sample_n_b[index]:
            print(f'Probability is bigger in N_a with the value {probability_sample_n_a[index]}')
        else:
            print(f'Probability is bigger in N_b with the value {probability_sample_n_b[index]}')

    # 2. Find val which has equal probability in both distributions.
    # Idea: find the intersection of two curves,
    # i.e. solve the quadratic equation:
    #  $a \cdot x^2 + b \cdot x + c = 0$ , where a, b and c are defined below

    n_a_var = n_a_std**2
```

```

n_b_var = n_b_std**2

a = 1 / (2 * n_a_var) - 1 / (2 * n_b_var)
b = n_b_mean / n_b_var - n_a_mean / n_a_var
c = n_a_mean ** 2 / (2 * n_a_var) - n_b_mean ** 2 / (2 * n_b_var) -
numpy.log(n_b_std / n_a_std)
intersection_points = numpy.roots([a, b, c])

print(f'Values with equal probabilities: {intersection_points}')

```

Ответ:

```

Value: 5
Probability is bigger in N_a with the value 0.24197072451914337
Value: 6
Probability is bigger in N_b with the value 0.08801633169107488
Value: 7
Probability is bigger in N_b with the value 0.09666702920071232
Values with equal probabilities: [5.7569948  1.70967186]

Process finished with exit code 0

```