Практические задания №1

6307 Ма Даньтин

Задание 1

```
#Предположим X и Y две случайные переменные отражающие возраст и вес, соотве
тственно. Рассмотрим случайную выборку из 20 наблюдений
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
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.

```
#Найти среднее
np.mean(X)
```

Ответ
71.45

```
#медиану
np.median(X)
```

Ответ
71.5

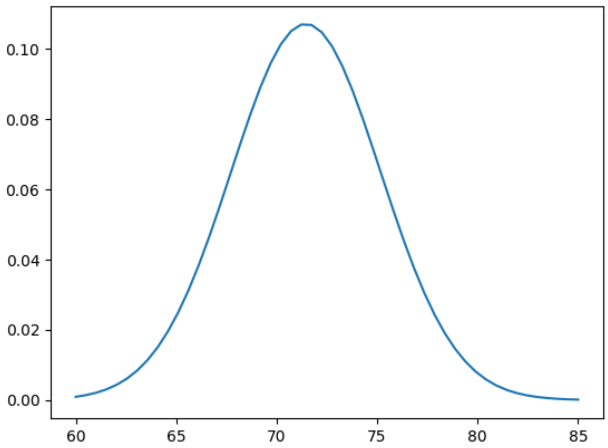
```
#моду величины
counts = np.bincount(X)
np.argmax(counts)
```

Ответ
74

```
#B. Найти дисперсию Y
np.var(Y)
```

Ответ
1369.2099999999998

```
#C. Построить график нормального распределения для X
ls = np.linspace(60, 85)
plt.plot(ls, stats.norm.pdf(ls, np.mean(X), np.std(X)))
plt.show()
```




```
#D. Найти вероятность того, что возраст больше 80
counts = np.sum(list(map(lambda x: x >= 80, X)))
P = counts/20
P
```

Ответ
0.0

```
#E. Найти двумерное мат. ожидания и ковариационную матрицу для этих двух вел
ичин
#ожидания
[np.mean(X), np.mean(Y)]
```

Ответ
[71.45, 164.7]

```
#ковариационную матрицу
np.cov([X,Y])
```

Ответ
array([[14.57631579, 128.87894737],
[128.87894737, 1441.27368421]])

```
np.corrcoef([X,Y])
```

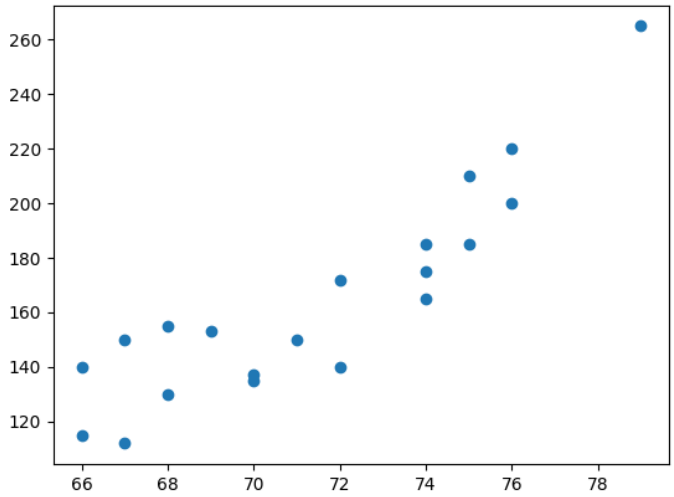
Ответ
array([[1. , 0.88917014],
[0.88917014, 1.]])

```
#F. Определять корреляцию между X и Y
np.corrcoef([X,Y])
```

Ответ

array([[1. , 0.88917014],
[0.88917014, 1.]])

```
#G. Построить диаграмму рассеяния, отображающая зависимость между возрастом и весом
plt.scatter(X,Y)
<matplotlib.collections.PathCollection object at 0x179BBA30>
plt.show()
```



Задание 2

```
#Для следующего набора данных
W = [[17, 17, 12], [11, 9, 13], [11, 8, 19]]
#Рассчитайте ковариационную матрицу
np.cov(W)
```

Ответ

array([[8.33333333, -5. , -15.83333333],
[-5. , 4. , 11.],
[-15.83333333, 11. , 32.33333333]])

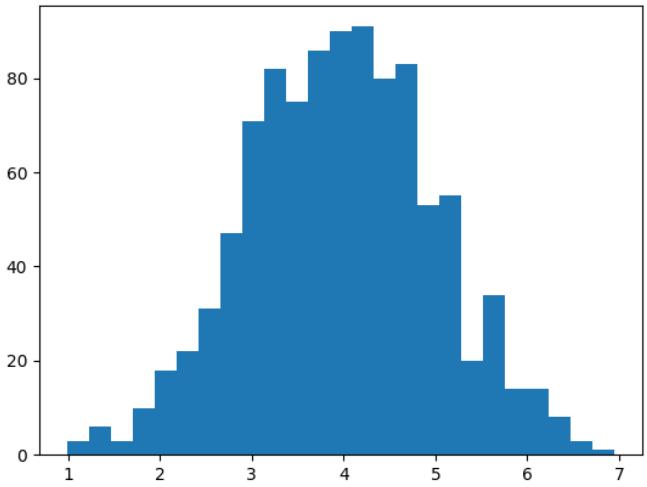
```
#обобщенную дисперсию
np.linalg.det(np.cov(W))
```

Ответ

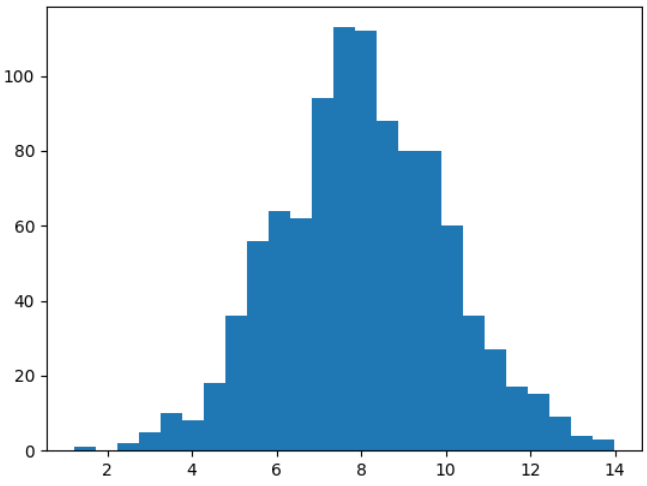
2.2484450948274876e-14

Задание 3

```
#Даны два одномерных нормальных распределения Na и Nb с мат. ожиданиями 4, 8 и СКО 1, 2 соответственно.
Na_mean =4
Nb_mean =8
Na_std = 1
Nb_std = 2
Na = np.random.normal(Na_mean, Na_std, 1000)
plt.figure(figsize=(20,10),dpi=100)
plt.hist(Na,25)
plt.show()
```




```
Nb = np.random.normal(Nb_mean, Nb_std, 1000)
plt.figure(figsize=(20,10),dpi=100)
plt.hist(Nb,25)
plt.show()
```



```
#Для каждого из значения {5,6,7} определите какое из распределений сгенериро
вало значение с большей вероятностью.
data = [5,6,7]
Pa = stats.norm.pdf(data, Na_mean, Na_std)
Pb = stats.norm.pdf(data, Nb_mean, Nb_std)
['Na' if difP > 0 else 'Nb' for difP in Pa - Pb]
```

Ответ
['Na', 'Nb', 'Nb']
5: распределений 'Na' сгенерировало значение с большей вероятностью
6: распределений 'Nb' сгенерировало значение с большей вероятностью
7: распределений 'Nb' сгенерировало значение с большей вероятностью

```
#Найди значение, которой могло быть сгенерировано обеими распределениями с р
авной вероятностью
x = np.linspace(5, 6, 100)
Pa = stats.norm.pdf(x, Na_mean, Na_std)
Pb = stats.norm.pdf(x, Nb_mean, Nb_std)
result = [x[i] for i in range(len(Pa)) if abs(Pa[i] - Pb[i]) <= 0.001]
result
```

Ответ
5.656565656565657