

practice 1

8 октября 2020 г.

```
[1]: import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

```
[2]: # 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 = [53, 175, 155, 135, 172, 150, 115, 137, 200, 130, 140, 265, 185, 112, 140,
      ↳150, 165, 185, 210, 220]
```

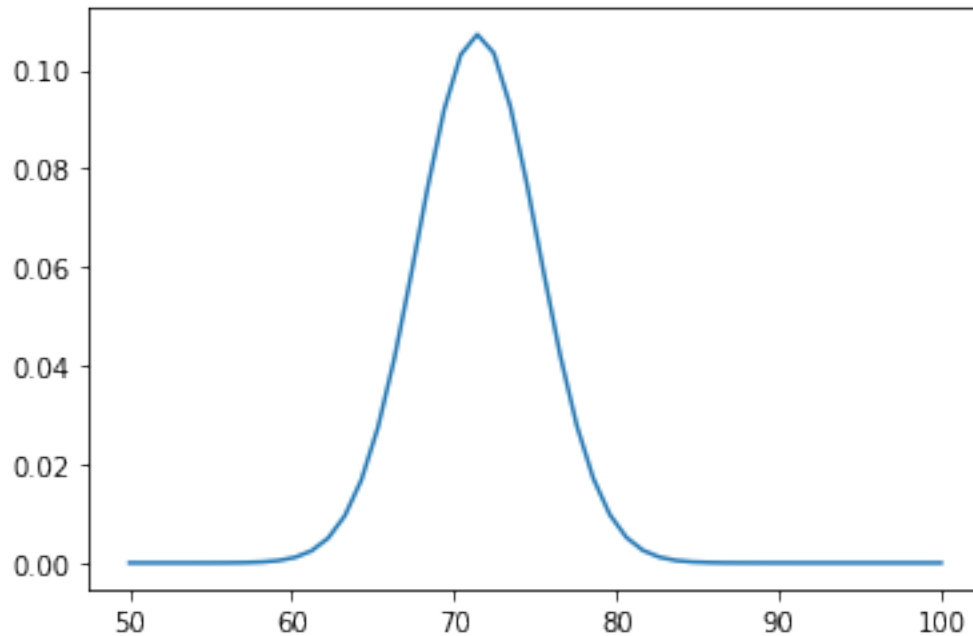
```
[3]: # A. Найти среднее, медиану и моду величины X
np.mean(X), np.median(X), *stats.mode(X)[0]
```

```
[3]: (71.45, 71.5, 74)
```

```
[4]: # B. Найти дисперсию Y
np.var(Y)
```

```
[4]: 1961.2100000000003
```

```
[26]: #C. Построить график нормального распределения для X
ls = np.linspace(50, 100)
plt.plot(ls, stats.norm.pdf(ls, np.mean(X), np.std(X)))
plt.show()
```



```
[25]: #D. Найти вероятность того, что возраст больше 80
z = (80 - np.mean(X)) / np.std(X)
S_0_z = 0.48 # по таблице при z = 2.3
P = 0.5 - S_0_z
P
```

```
[25]: 0.0200000000000000018
```

```
[18]: #E. Найти двумерное мат. ожидание
[np.mean(X), np.mean(Y)]
```

```
[18]: [71.45, 159.7]
```

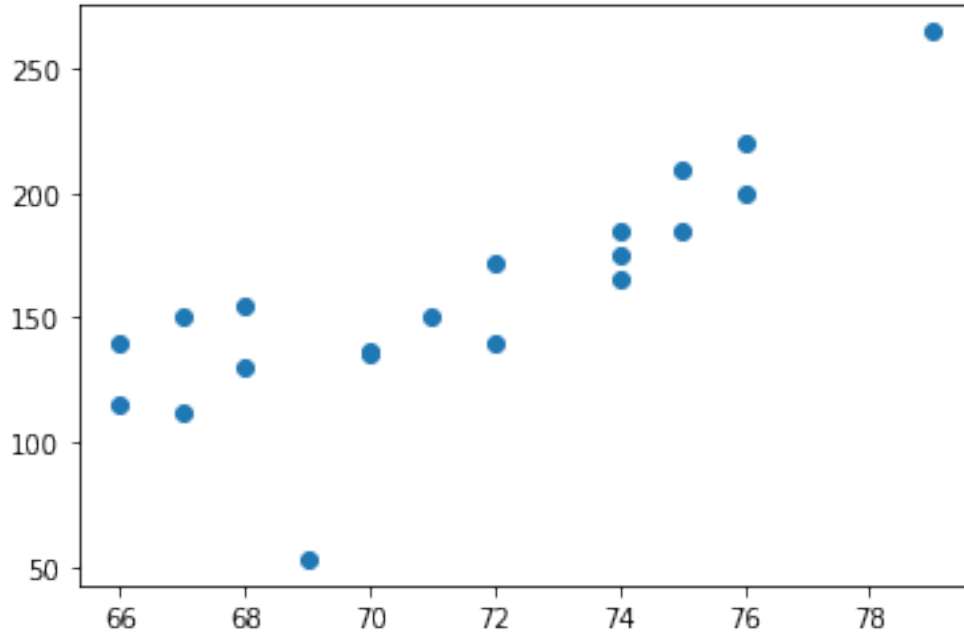
```
[8]: # и ковариационную матрицу для этих двух величин
np.cov(X, Y)
```

```
[8]: array([[ 14.57631579, 141.77368421],
          [ 141.77368421, 2064.43157895]])
```

```
[9]: # F. Определить корреляцию между X и Y
np.corrcoef(X, Y)[0, 1]
```

```
[9]: 0.8172811723193554
```

```
[10]: # G. Построить диаграмму рассеяния, отображающая зависимость между возрастом и
      ↪ весом
      plt.scatter(X, Y)
      plt.show()
```



```
[11]: # 2. Для следующего набора данных
      m = [[17, 17, 12],
           [11, 9, 13],
           [11, 8, 19]]
```

```
[12]: # рассчитайте ковариационную матрицу
      cov_m = np.cov(m)
      cov_m
```

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

```
[27]: # и обобщенную дисперсию
      np.linalg.det(cov_m)
```

```
[27]: 2.2204460492503156e-14
```

```
[14]: # 3. Даны два одномерных нормальных распределения Na и Nb с мат. ожиданиями 4, 8
      ↪ и СКО 1, 2 соответственно.
```

```
Na = stats.norm(4, 1)
Nb = stats.norm(8, 2)
```

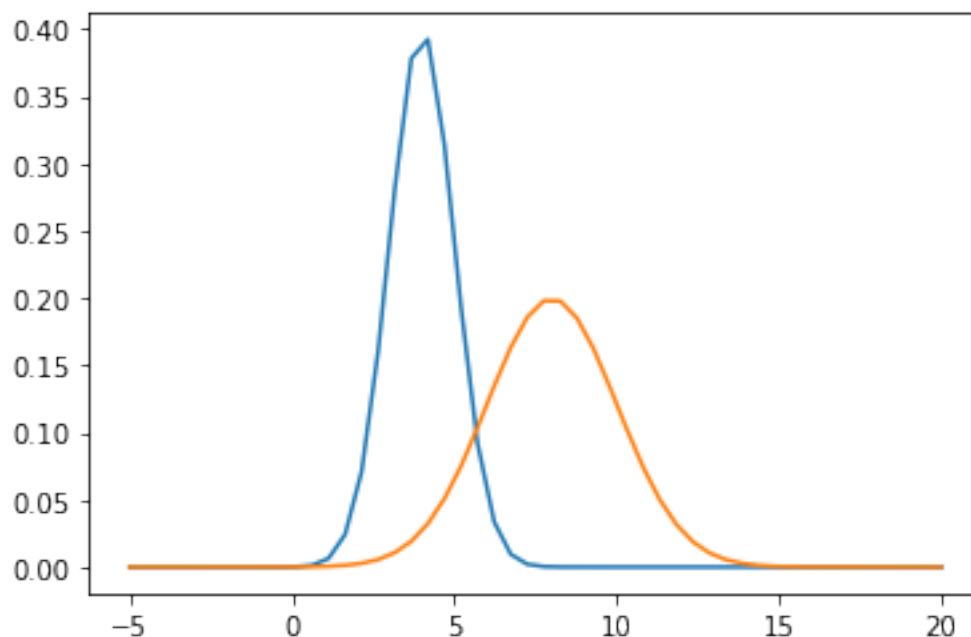
```
[15]: # A. Для каждого из значения {5,6,7} определите какое из распределений
      ↪ сгенерировало значение с большей вероятностью.
```

```
data = [5, 6, 7]
Pa = Na.pdf(data)
Pb = Nb.pdf(data)
['Na' if difP > 0 else 'Nb' for difP in Pa - Pb]
```

```
[15]: ['Na', 'Nb', 'Nb']
```

```
[16]: #B. Найди значение, которой могло быть сгенерировано обеими распределениями с
      ↪ равной вероятностью
```

```
# 1) График - пересечение
ls = np.linspace(-5, 20)
plt.plot(ls, Na.pdf(ls))
plt.plot(ls, Nb.pdf(ls))
plt.show()
```



```
[28]: # 2) Нахождение
```

```
ls = np.linspace(5, 6, 100)
Pa = Na.pdf(ls)
Pb = Nb.pdf(ls)
[ls[i] for i in range(len(Pa)) if abs(Pa[i] - Pb[i]) <= 0.001]
```

[28] : [5.656565656565657]