

```
In [20]: import numpy as np
```

1

Даны значения величины заработной платы заемщиков банка (zp) и значения их поведенческого кредитного скоринга (ks): zp = [35, 45, 190, 200, 40, 70, 54, 150, 120, 110], ks = [401, 574, 874, 919, 459, 739, 653, 902, 746, 832]. Найдите ковариацию этих двух величин с помощью элементарных действий, а затем с помощью функции cov из numpy. Полученные значения должны быть равны коэффициенту корреляции Пирсона с помощью ковариации и среднеквадратичных отклонений двух признаков.

```
In [21]: zp = np.array([35, 45, 190, 200, 40, 70, 54, 150, 120, 110])
ks = np.array([401, 574, 874, 919, 459, 739, 653, 902, 746, 832])

cov = (zp*ks).mean() - zp.mean()*ks.mean()

print("Ковариация =", cov)
print("np.cov:", np.cov(zp, ks, ddof=0)[0][1])
```

```
Ковариация = 9157.839999999997
np.cov: 9157.840000000002
```

```
In [22]: r = cov / (zp.std() * ks.std())
print('Коэффициент корреляции Пирсона =', r)
print('np.corrcoef:', np.corrcoef(zp, ks)[0][1])
```

```
Коэффициент корреляции Пирсона = 0.8874900920739158
np.corrcoef: 0.8874900920739163
```

2

Измерены значения IQ выборки студентов, обучающихся в местных технических вузах: 131, 125, 115, 122, 131, 115, 107, 99, 125, 111. Известно, что в генеральной совокупности IQ распределен нормально. Найдите доверительный интервал для математического ожидания с надежностью 0.95.

```
In [36]: iq = [131, 125, 115, 122, 131, 115, 107, 99, 125, 111]
a = 0.05
n = len(iq)
mu = np.array(iq).mean()
std = np.array(iq).std()
t = 2.262 # t от a/2
print('Доверительный интервал, описывающий 95% выборки:\n',
      [mu-t*std/np.sqrt(n), mu+t*std/np.sqrt(n)])
```

```
Доверительный интервал, описывающий 95% выборки:
[110.94370977419166, 125.25629022580833]
```

3

Известно, что рост футболистов в сборной распределен нормально с дисперсией генеральной совокупности, равной 25 кв.см. Объем выборки равен 27, среднее выборочное составляет 174.2. Найдите доверительный интервал для математического ожидания с надежностью 0.95.

```
In [38]: d = 25; n = 27; mu = 174.2
std = np.sqrt(d)
Z = 1.96
print('Доверительный интервал для математического ожидания с надежностью 95%:',
      [mu-Z*std/np.sqrt(n), mu+Z*std/np.sqrt(n)])
```

```
Доверительный интервал для математического ожидания с надежностью 95%: [172.31398912064722, 176.08601087935276]
```