

## Практическое задание 6

In [1]:

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import warnings
5 import scipy
6 from scipy import stats
7 warnings.filterwarnings('ignore')
```

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. Найдите коэффициент корреляции Пирсона с помощью ковариации и среднеквадратичных отклонений двух признаков, а затем с использованием функций из библиотек numpy и pandas.

In [2]:

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

In [3]:

```
1 cov = np.mean(zp * ks) - np.mean(zp)*np.mean(ks)
2 cov
```

Out[3]:

9157.839999999997

In [4]:

```
1 np.cov(zp, ks, ddof = 0)
```

Out[4]:

```
array([[ 3494.64,  9157.84],
       [ 9157.84, 30468.89]])
```

Видим, что значения ковариаций равны.  
Найдем теперь коэффициент корреляции

In [5]:

```
1 coef_corr = cov/(np.std(zp)*np.std(ks))
2 coef_corr
```

Out[5]:

0.8874900920739158

In [6]:

```
1 np.corrcoef(zp, ks)
```

Out[6]:

```
array([[1., 0.88749009],
       [0.88749009, 1.]])
```

Видим, что значения корреляций также равны

**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.**

Поскольку СКО генеральной совокупности нам не известно, воспользуемся критерием Стьюдента. Формула поиска доверительного интервала немного изменится:

$$\bar{X} \pm t_{\frac{\alpha}{2}} \cdot \frac{S}{\sqrt{n}}$$

Найдем  $\bar{X}$ , S и n из выборки

In [7]:

```
1 IQ = pd.DataFrame([131, 125, 115, 122, 131, 115, 107, 99, 125, 111])
```

In [8]:

```
1 S = np.sqrt(((IQ[0] - IQ[0].mean())**2).sum() / (IQ[0].count() - 1))
2 S
```

Out[8]:

10.54566788359614

In [9]:

```
1 n = IQ[0].count()
2 n
```

Out[9]:

10

In [10]:

```
1 X = IQ[0].mean()
2 X
```

Out[10]:

118.1

Все данные у нас есть, кроме t-критерия. Чтобы долго не искать его в таблице, найдем его используя средства питона.

In [11]:

```
1 t_value = scipy.stats.t.ppf(0.975, 10-1)
2 t_value
```

Out[11]:

2.2621571627409915

In [12]:

```
1 interval = t_value*(S/np.sqrt(10))
2 interval
```

Out[12]:

7.543916348412745

In [13]:

```
1 print("Доверительный интервал: ", (X - interval, X + interval))
```

Доверительный интервал: (110.55608365158724, 125.64391634841274)

**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.**

Поскольку мы имеем дело с нормальным распределением с известным среднеквадратичным отклонением, то воспользуемся формулой:

$$\bar{X} \pm Z_{\frac{\alpha}{2}} \cdot \frac{\sigma}{\sqrt{n}}$$

Все данные у нас есть, кроме Z-значения. Чтобы долго не искать его в таблице, найдем его используя средства питона.

In [14]:

```
1 z_value = stats.norm.ppf(q = 0.975)
2 z_value
```

Out[14]:

1.959963984540054

In [15]:

```
1 interval = z_value*(5/np.sqrt(27))
2 interval
```

Out[15]:

1.8859762234602861

In [16]:

```
1 print("Доверительный интервал: ", (174.2 - interval, 174.2 + interval))
```

Доверительный интервал: (172.3140237765397, 176.08597622346028)