Теория вероятностей и математическая статистика

Урок 6. Взаимосвязь величин. Параметрические и непараметрические показатели корреляции. Корреляционный анализ.

Урок 6.

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.
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.
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 альфа =0,95

B [37]:
```
import numpy as np
from statsmodels.stats.weightstats import _tconfint_generic as t_stat  # t-критерий
from statsmodels.stats.weightstats import _zconfint_generic as z_stat  # z-критерий
```

Задача 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.

B [2]:
```
# s
zp = np.array([35, 45, 190, 200, 40, 70, 54, 150, 120, 110])
zp
```

Out[2]:
```
array([ 35,  45, 190, 200,  40,  70,  54, 150, 120, 110])
```

B [3]:
```
# p
ks = np.array([401, 574, 874, 919, 459, 739, 653, 902, 746, 832])
ks
```

Out[3]:
```
array([401, 574, 874, 919, 459, 739, 653, 902, 746, 832])
```

B [4]:
```
# Ковариация cov = M(XY) - M(X)M(Y)
cov_ks = np.mean(ks*zp) - np.mean(ks) * np.mean(zp)
cov_ks
```

Out[4]:
```
9157.839999999997
```

B [5]:
```
# Ковариация смещённая (ddof=0)
np.cov(zp, ks, ddof=0)
# np.cov(ks,zp, ddof=0)
```

Out[5]:
```
array([[ 3494.64,  9157.84],
       [ 9157.84, 30468.89]])
```

B [6]:
```
# Ковариация несмещённая (ddof=1)
np.cov(ks,zp)
```

Out[6]:
```
array([[33854.32222222, 10175.37777778],
       [10175.37777778,  3882.93333333]])
```

B [7]:
```
# Ковариация несмещённая
np.cov(ks, zp, ddof=1)
```

Out[7]:
```
array([[33854.32222222, 10175.37777778],
       [10175.37777778,  3882.93333333]])
```

B [8]:
```
# Стандартное отклонение смещённое
std_ks = np.std(ks, ddof=0)
# std_ks = ks.std()
std_ks
```

Out[8]:
```
174.55340157098058
```

В [9]:
```
# Стандартное отклонение смещённое
std_zp = np.std(zp, ddof=0)
#std_zp = zp.std()
std_zp
```

Out[9]:
```
59.115480206118605
```

В [10]:
```
# Коэффициент корреляции Пирсона 9157.839999999997/(174.55340157098058*59.115480206118605)
korr_ks = cov_ks/(std_ks*std_zp)
korr_ks
```

Out[10]:
```
0.8874900920739158
```

В [11]:
```
# Коэффициент корреляции Пирсона
np.corrcoef(ks,zp)
```

Out[11]:
```
array([[1.        , 0.88749009],
       [0.88749009, 1.        ]])
```

Включите ddof=1 если вы вычисляете np.std() для образца, взятого из вашего полного набора данных.

Убедитесь, что ddof=0 если вы вычисляете np.std() для всей совокупности

Задача 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.

σ не известно. Используем t-критерий

В [12]:
```
X = np.array([131.0, 125.0, 115.0, 122.0, 131.0, 115.0, 107.0, 99.0, 125.0, 111.0])
X
```

Out[12]:
```
array([131., 125., 115., 122., 131., 115., 107.,  99., 125., 111.])
```

В [13]:
```
n=len(X)  # Объём выборки
X_cp = X.mean()  # Среднее арифмитическое для данной выборки
nu = n-1  # Число степеней свободы
alpha = 1 - 0.95
t_975_9 = 2.262  # Коэффициент Стьюдента табличный t(p=0.975, nu=9)
#sigma = X.std() # Стандартное отклонение смещённое
sigma = X.std(ddof=1)  # Несмещённое среднеквадратичное отклонение
print(f'Объём выборки n = {n}')
print(f'Число степеней свободы nu = {nu}')
print(f'Среднее выборочное X_cp = {X_cp}')
print(f'Статистический уровень значимости alpha = {alpha}')
print(f'Коэффициент Стьюдента табличный t(p=0.975, nu=9) = {t_975_9}')
# print(f'Смещённое среднеквадратичное отклонение sigma = {sigma}')
print(f'Несмещённое среднеквадратичное отклонение sigma = {sigma}')
```

```
Объём выборки n = 10
Число степеней свободы nu = 9
Среднее выборочное X_cp = 118.1
Статистический уровень значимости alpha = 0.050000000000000044
Коэффициент Стьюдента табличный t(p=0.975, nu=9) = 2.262
Несмещённое среднеквадратичное отклонение sigma = 10.54566788359614
```

Доверительный интервал [110.5566; 125.6434] с вероятностью 95%

$$\overline{X} \pm t_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}} = 118.1 \pm 2.262 \cdot \frac{10.5567}{\sqrt{10}} = 118.1 \pm 7.5434$$

В [14]:
```
se=sigma/np.sqrt(n)
se
```

Out[14]:
```
3.3348329959851224
```

В [15]:
```
mean_std_X = t_975_9*sigma/np.sqrt(n)
mean_std_X
```

Out[15]:
```
7.543392236918348
```

В [16]:
```
X_cp - t_975_9*sigma/np.sqrt(n)
```

Out[16]:
```
110.55660776308164
```

В [17]:
```
X_cp + t_975_9*sigma/np.sqrt(n)
```

Out[17]:
```
125.64339223691834
```

В [18]:
```
t_stat(X_cp, sigma/np.sqrt(n), n - 1, 0.05, 'two-sided')
```

Out[18]:
```
(110.55608365158724, 125.64391634841274)
```

Ответ: Доверительный интервал [110.5566; 125.6434] с вероятностью 95%

Задача 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 альфа =0,95

σ известно. Используем z-критерий.

В [19]:
```
M =  174.2  # выборочная средняя
D = 25.0  # дисперсией генеральной совокупности
n = 27  # объем выборки
sigma = np.sqrt(D)  # средне квадратическое отклонением
alpha = 0.05  # статистический уровень значимости α=0.05
```

В [20]:
```
print(f'Объём выборки n = {n}')
print(f'Среднее выборочное X_cp = {X_cp}')
print(f'дисперсией генеральной совокупности D = {D}')
print(f'Среднеквадратичное отклонение sigma = {sigma}')
print(f'Статистический уровень значимости alpha = {alpha}')
```

```
Объём выборки n = 27
Среднее выборочное X_cp = 118.1
дисперсией генеральной совокупности D = 25.0
Среднеквадратичное отклонение sigma = 5.0
Статистический уровень значимости alpha = 0.05
```

z табличное для $\alpha/2 = 0.025$

$$z_{\alpha/2} = 1.96$$

В [21]:
```
z_alpfa_2=1.96  # z табличное для  α/2
```

Доверительный интервал $[172.3140; 176.0860]$ с вероятностью 95%

$$\overline{X} \pm z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}} = 174.2 \pm 1.96 \cdot \frac{5.0}{\sqrt{27}} \approx 174.2 \pm 1.96 * 0.96225 \approx 174.2 \pm 1.8860$$

В [22]:
```
se=sigma/np.sqrt(n)
se
```

Out[22]:
```
0.9622504486493763
```

В [33]:
```
se*z_alpfa_2
```

Out[33]:
```
1.8860108793527774
```

В [34]:
```
M - z_alpfa_2*sigma/np.sqrt(n)
```

Out[34]:
```
172.31398912064722
```

В [35]:
```
M + z_alpfa_2*sigma/np.sqrt(n)
```

Out[35]:
```
176.08601087935276
```

В [36]:
```
# return _zconfint_generic(self.mean, self.std_mean, alpha, alternative)
z_stat(M, se, alpha, 'two-sided')
```

Out[36]:
```
(172.3140237765397, 176.08597622346028)
```

Ответ: Доверительный интервал [172.3140237765397, 176.08597622346028] с вероятностью 95%

В [ ]: