

Как я использовала бутстрэп для построения доверительного интервала для показателя *Corr.*

В качестве данных для эксперимента были взяты два показателя за 2021 год по 81 региону РФ с сайта Росстата:

Y – доля продаж через Интернет в общем объеме оборота розничной торговли

X – удельный вес городского населения в общей численности населения

Данные импортировала в Python из Excel.

	y_online_trade	city
0	3.5	67.7
1	1.7	70.6
2	1.9	78.2
3	5.0	68.1
4	3.2	81.9
...
76	2.2	82.2
77	2.4	68.1
78	1.3	96.1
79	1.2	82.7
80	1.0	68.1

81 rows × 2 columns

Посчитала корреляцию точно, результат 0,39:

```
def corr(Y, X):  
    corr_mat = np.corrcoef(Y, X)  
    return corr_mat[0, 1]
```

```
corr(Y, X)
```

```
0.3888443812594289
```

Для того, чтобы у нас при генерации выборки не была потеряна связь между значениями по регионам, сгенерировала сначала номера, а затем уже с привязкой к ним, выборки по Y и X.

```
rng = np.random.default_rng(111111)
rand_numb = rng.choice(range(len(X)), size=len(X))
rand_numb

array([61, 28, 45, 53, 34, 72, 17, 45, 1, 33, 56, 23, 17, 69, 50, 55, 38,
       5, 68, 80, 38, 72, 23, 77, 76, 20, 49, 12, 60, 1, 75, 12, 8, 25,
      13, 29, 47, 38, 13, 69, 52, 64, 64, 25, 18, 19, 38, 56, 10, 22, 1,
      48, 37, 71, 41, 31, 3, 8, 11, 64, 49, 10, 63, 20, 57, 77, 49, 63,
      58, 78, 53, 36, 47, 8, 32, 38, 73, 32, 8, 55, 19])
```

```
X_1 = X[rand_numb]
X_1

61    29.1
28    50.7
45    66.3
53    75.9
34    45.3
...
73    68.7
32    68.1
8     64.6
55    81.9
19    78.6
Name: city, Length: 81, dtype: float64
```

```
Y_1 = Y[rand_numb]
Y_1

61    3.8
28    5.9
45    1.9
53    4.6
34    0.1
...
73    0.9
32    6.2
```

В выборке $\text{Corr} = 0,41$

```
corr(Y_1, X_1)

0.41248073772940097
```

И теперь магия бутстрэпа и получаем доверительный интервал для Corr [0.13; 0.59]

```
boot_YX = IIDBootstrap(Y, X, seed=111111)
boot_YX.conf_int(corr, method='percentile', reps=10000, size=0.95)

array([[0.12759727],
       [0.59271458]])
```