

Как я использовала бутстрэп (bootstrap)?

Для использования метода оценки стандартных отклонений и нахождения доверительных интервалов статистических функционалов, я выбрала набор данных, состоящий из 50 наблюдений по скорости (в м/ч) и длине тормозного пути (в футах) автомобилей, записанный в 1920-х годах. Чтобы получить вышеуказанный набор данных, я воспользовалась командой, которая позволяет импортировать данные, которые встроены в R:

```
!pip install rdatasets
from rdatasets import data
```

Из всего набора данных по автомобилям я решила использовать выборку по скорости для применения бутстрэпа. С помощью команд **np.mean(speed)** и **np.var(speed)** я нашла значения $E(\text{Speed}_i) = 15.5$ и $\text{Var}(\text{Speed}_i) \approx 5.349$. Таким образом, средняя скорость автомобиля равна 15,5 м/ч, а значение дисперсии говорит о том, что данные по скорости автомобилей, в среднем отклоняются от среднего значения на 5,349 единиц.

На основе вышеуказанных значений, я сгенерировала новую выборку с помощью команды:

```
rng = np.random.default_rng(111111)
speed_1 = rng.normal(loc=speed_mean, scale=speed_var_1, size=50)
```

Я получила выборочное среднее (команда **np.mean(speed_1)**) и точечную оценку для математического ожидания (команда $\text{mu_hat} = \text{np.mean}(\text{speed_1})$) равное 15,45.

Для нахождения доверительного интервала, необходимо превратить точечную оценку в интервальную с помощью команды:

```
speed_2 = rng.choice(speed_1, size=len(speed_1))
```

В результате я получила случайную выборку из вышеуказанных данных по скорости автомобилей, которая дает возможность создать новую бутстрэп точечную оценку для математического ожидания. Команда **np.mean(speed_2)** дала результат по выборочному среднему случайной выборки равный 15,31.

Для создания 100 новых выборок я использовала команду:

```
n_boot = 100
mu_hat_speed = [np.mean(rng.choice(speed_1, size=len(speed_1))) for i in range(n_boot)]
```

Следующим действием я построила 95% доверительный интервал для математического ожидания с помощью команды:

```
[np.quantile(mu_hat_speed, 0.025), np.quantile(mu_hat_speed, 0.975)]
```

Таким образом, истинное значение математического ожидания находится в интервале [14.19, 16.73], также данный интервал покрывает вышеуказанные оцененные параметры.

```

import numpy as np
import pandas as pd
import seaborn as sns
import math

!pip install arch

from arch.bootstrap import IIDBootstrap, IndependentSamplesBootstrap

!pip install rdatasets
from rdatasets import data

x = data('cars')

speed = [4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12, 12, 12, 12, 13, 13, 13, 13,
, 14, 14, 14, 14, 15, 15, 15, 16, 16, 17, 17, 17, 18, 18, 18, 18, 19, 19, 20,
20, 20, 20, 22, 23, 24, 24, 24, 24, 24, 25]

speed_mean = np.mean(speed)

speed_mean

15.5

speed_var = np.var(speed)

speed_var

28.61

speed_var_1 = math.sqrt(speed_var)
speed_var_1

5.348831648126533

rng = np.random.default_rng(111111)
speed_1 = rng.normal(loc=speed_mean, scale=speed_var_1, size=50)

speed_1

array([20.51527986, 11.55070293, 16.60235152, 12.58399659, 13.56682304,
14.65084448 , 23.44588311, 17.53845708, 20.48397354, 20.28394024,
14.80135786, 20.592476 , 15.59682187, 13.54831715, 16.15352403,
19.08770422, 12.06619963, 4.25080091, 20.56799168, 6.45412373,
20.84639519, 10.58547711, 24.9104054 , 13.69210874, 16.51928943,
5.64001912, 15.31133473, 14.71349304, 18.65029751, 11.78566881,
15.72687889, 15.45345559, 19.65341276, 19.73606821, 6.45889417,
8.45715032, 30.30525296, 11.09956175, 13.80176862, 16.93839615,
15.97839835, 11.25690271, 9.4665812 , 19.80094879, 14.01450656,
16.15311539, 17.0656341 , 17.3752673 , 14.21069112, 12.68786419])

np.mean(speed_1)

15.45273615997394

mu_hat = np.mean(speed_1)
mu_hat

15.45273615997394

speed_2 = rng.choice(speed_1, size=len(speed_1))

```

```
speed_2
```

```
array([12.06619963, 16.15352403, 19.73606821, 12.58399659, 15.72687889,  
       6.45889417, 20.592476 , 14.6508448 , 14.01450656, 19.08770422,  
       10.58547711, 19.80094879, 13.69210874, 6.45412373, 16.51928943,  
       16.51928943, 20.48397354, 14.01450656, 17.3752673 , 24.9104054 ,  
       18.65029751, 20.51527986, 11.25690271, 6.45412373, 20.84639519,  
       15.97839835, 13.69210874, 17.0656341 , 11.55070293, 11.78566881,  
       20.28394024, 6.45889417, 16.51928943, 14.21069112, 16.93839615,  
       19.73606821, 17.0656341 , 12.06619963, 11.78566881, 10.58547711,  
       4.25080091, 18.65029751, 17.3752673 , 16.93839615, 19.73606821,  
       20.48397354, 14.80135786, 17.3752673 , 19.08770422, 11.78566881])
```

```
np.mean(speed_2)
```

```
15.307141116564267
```

```
n_boot = 100
```

```
mu_hat_speed = [np.mean(rng.choice(speed_1, size=len(speed_1))) for i in range(n_boot)]
```

```
mu_hat_speed
```

```
[15.886808612899152,  
 14.478524561292502,  
 16.20798165033302,  
 15.764327602481044,  
 15.67342803841248,  
 ...,  
 16.049678480586167]
```

```
[np.quantile(mu_hat_speed, 0.025), np.quantile(mu_hat_speed, 0.975)]
```

```
[14.19442909391508, 16.732317749184908]
```