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

```
!pip install arch
```

```
Collecting arch
  Downloading arch-6.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (981 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 981.7/981.7 kB 13.4 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.19 in /usr/local/lib/python3.10/dist-packages (from arch) (1.23.5)
Requirement already satisfied: scipy>=1.5 in /usr/local/lib/python3.10/dist-packages (from arch) (1.11.3)
Requirement already satisfied: pandas>=1.1 in /usr/local/lib/python3.10/dist-packages (from arch) (1.5.3)
Requirement already satisfied: statsmodels>=0.12 in /usr/local/lib/python3.10/dist-packages (from arch) (0.14.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1->arch) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1->arch) (2023.3.post1)
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.12->arch) (0.5.3)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.12->arch) (23.2)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5.2->statsmodels>=0.12->arch) (1.16.0)
Installing collected packages: arch
Successfully installed arch-6.2.0
```

```
from arch.bootstrap import IIDBootstrap, IndependentSamplesBootstrap
```

```
rng = np.random.default_rng(365)
x = rng.normal(loc=450, scale=4, size=10)
x
```

```
array([449.45436404, 450.37657724, 449.74798231, 454.21566562,
       447.25327281, 447.96311308, 446.8001896 , 452.92167302,
       450.35301754, 441.52896959])
```

Моя цель состояла в оценке среднего курса доллара к тенге за последний год. Я взял среднее значение 450 тенге, отклонение 4, выборка 10 дней

```
np.mean(x)
```

```
449.06148248467406
```

Получил среднее значение 449.06148248467406

Чтобы изменить содержимое ячейки, дважды нажмите на нее (или выберите "Ввод")

```
mu_hat = np.mean(x)
mu_hat
```

```
449.06148248467406
```

```
x_star1 = rng.choice(x, size=len(x))
```

```
тут сделал случайную выборку, с размером 10 наблюдений
```

```
x_star1
```

```
array([447.25327281, 450.35301754, 446.8001896 , 447.25327281,
       441.52896959, 449.45436404, 452.92167302, 446.8001896 ,
       450.37657724, 454.21566562])
```

```
np.mean(x_star1)
```

```
448.69571918624786
```

Получил среднее значение 449.69571918624786

```
x_star2 = rng.choice(x, size=len(x))
np.mean(x_star2)
```

```
    449.353300160278
```

```
x_star2
```

```
    array([450.37657724, 449.74798231, 447.25327281, 447.25327281,
           452.92167302, 447.25327281, 449.74798231, 446.8001896 ,
           454.21566562, 447.96311308])
```

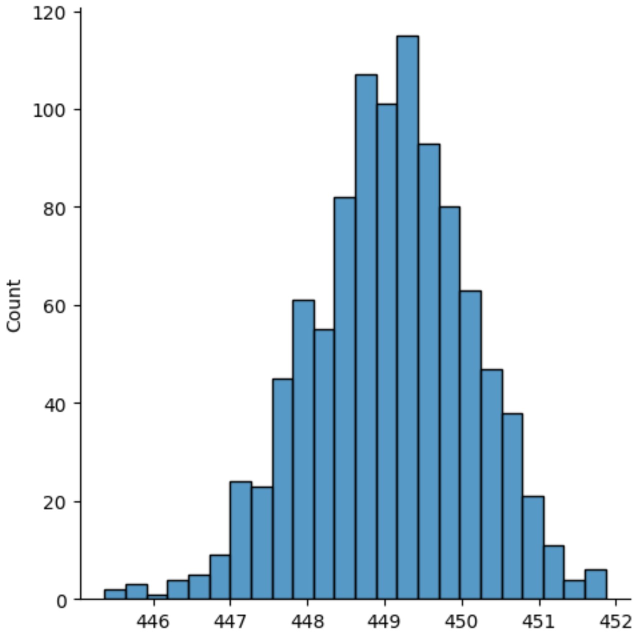
```
n_boot = 1000
mu_hat_star = [np.mean(rng.choice(x, size=len(x))) for i in range(n_boot)]
```

```
mu_hat_star[1:10]
```

```
    [448.0897911986951,
     449.9008417080152,
     450.0246218733998,
     449.7022856644877,
     448.39760651688175,
     448.1399652958847,
     448.790902101539,
     449.6539049905212,
     448.01851230581525]
```

```
sns.displot(x=np.array(mu_hat_star))
```

```
    <seaborn.axisgrid.FacetGrid at 0x784c31a4cb20>
```



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

```
    [447.03164407567215, 450.9493084386595]
```

Отсекаем вверхние и нижние 0,25 и получаем интервал 447.03 и 450.949 с доверительным интервалом в 95%