

▼ Бутстреп-анализ

Применение метода бутстрепа для оценки доверительного интервала среднего значения на примере набора данных результатов экзаменов студентов.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

▼ Генерация данных и функция бутстрепа

```
# Задаем seed для воспроизводимости результатов
np.random.seed(42)

# Создаем набор данных: результаты экзамена для 30 студентов
exam_scores = np.random.randint(50, 100, 30)

# Функция для бутстрепа
def bootstrap(data, n_bootstrap):
    bootstrap_means = []
    for _ in range(n_bootstrap):
        # Выбираем выборки с возвращением и вычисляем среднее
        sample = np.random.choice(data, size=len(data), replace=True)
        bootstrap_means.append(np.mean(sample))
    return bootstrap_means
```

▼ Выполнение бутстреп-анализа

```
# Применяем бутстреп (1000 повторений)
bootstrap_results = bootstrap(exam_scores, 1000)

# Вычисляем 95% доверительный интервал
lower_bound = np.percentile(bootstrap_results, 2.5)
upper_bound = np.percentile(bootstrap_results, 97.5)
lower_bound, upper_bound

(69.56583333333333, 78.5675)
```

▼ Визуализация результатов

```
# Визуализация распределения средних значений бутстреп-выборок
plt.figure(figsize=(10, 6))
sns.histplot(bootstrap_results, kde=True, color='blue')
plt.axvline(x=lower_bound, color='red', linestyle='--', label='2.5% percentile (Lower Bound)')
plt.axvline(x=upper_bound, color='green', linestyle='--', label='97.5% percentile (Upper Bound)')
plt.title('Bootstrap Distribution of Sample Means')
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

