

```

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import norm

# Загрузка данных
df = pd.read_csv('sales.csv')

# Разделение данных на обучающую и тестовую выборки
train_size = int(0.7 * len(df))
train, test = df.iloc[:train_size], df.iloc[train_size:]

# Оценка среднего значения выручки с помощью бутстрэпа
def bootstrap_mean(data):
    bootstrap_samples = bootstrap.Bootstrap(data, n_bootstraps=1000, sampling=bootstrap.ALL)
    bootstrap_means = bootstrap_samples.sample_with_replacement().statistic.apply(np.mean)
    mean_estimate = bootstrap_means.mean()
    stderr = bootstrap_means.std()
    return mean_estimate, stderr

mean_est, stderr = bootstrap_mean([train['revenue']])
print('Mean estimate:', mean_est)
print('Standard error:', stderr)

```

Данные были взяты из файла sales.csv. Затем данные делятся на обучающую и тестовую выборки.

Обучающая выборка составляет 70% от всех данных.

Функция `bootstrap_mean` принимает данные (в данном случае это выручка в обучающей выборке) и использует их для оценки среднего значения с помощью метода бутстрэп. Для этого создается несколько случайных выборок (`bootstrap_samples`) из исходных данных. Для каждой выборки вычисляется среднее значение (`bootstrap_means`). Затем вычисляется среднее значение этих средних значений (`mean_estimate`) и стандартное отклонение (`stderr`).

В конце функция возвращает оценку среднего значения (`mean_est`) и стандартную ошибку (`stderr`)

Для оценки использовались следующие параметры:

- среднее значение выручки
- стандартное отклонение выручки
- коэффициент корреляции между количеством проданных товаров и выручкой.

Вывод: Bootstrap позволил получить более точную оценку параметров распределения выручки и более надежные интервальные оценки по сравнению с классической оценкой.