

# АВ тестирование

## А/А тест для проверки

### Подготовка данных

Изучим данные, уберем выбросы.

Рассматриваем 2 датасета, сформированных за 3 дня. Первый - 3 дня до эксперимента, использовался для валидации. Второй - данные с запущенным экспериментом.

```
In [30]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [38]: def prepare_data(filename):
    data = pd.read_csv(filename, names=["id", "clicks", "in_group"])
    print(data.head())

    # Смотрим персентили
    print(f"100 percentile: {np.percentile(data['clicks'], 100)}")
    perc_95 = np.percentile(data['clicks'], 95)
    print(f"95 percentile: {perc_95}")
    print(f"85 percentile: {np.percentile(data['clicks'], 85)}")

    # Фильтруем данные по 95 персентиллю, таким образом убираем выбросы
    data = data.loc[data['clicks'] <= perc_95]

    # Средние:
    a = data[data['in_group']]
    b = data[data['in_group'] != True]
    print(f"Среднее в контрольной группе {np.mean(a['clicks'])}")
    print(f"Среднее в тестовой группе {np.mean(b['clicks'])}")

    # Гистограмма
    plt.figure(figsize=(16, 9))
    plt.title('Гистограмма количества кликов')
    plt.hist(data['clicks'], density=True)
    plt.xlabel('Количество кликов')
    plt.ylabel('Количество сессий') # Поскольку данные за несколько дней, на каждый
    plt.legend()
    plt.show()

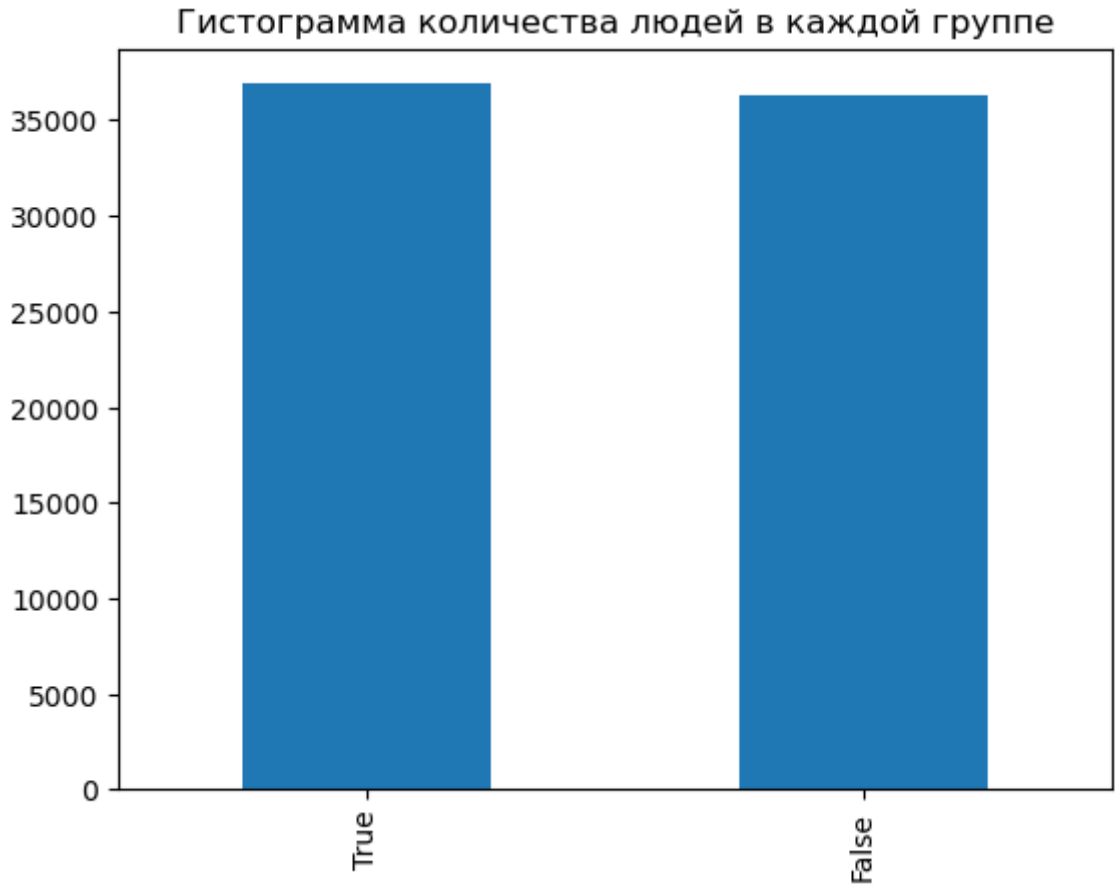
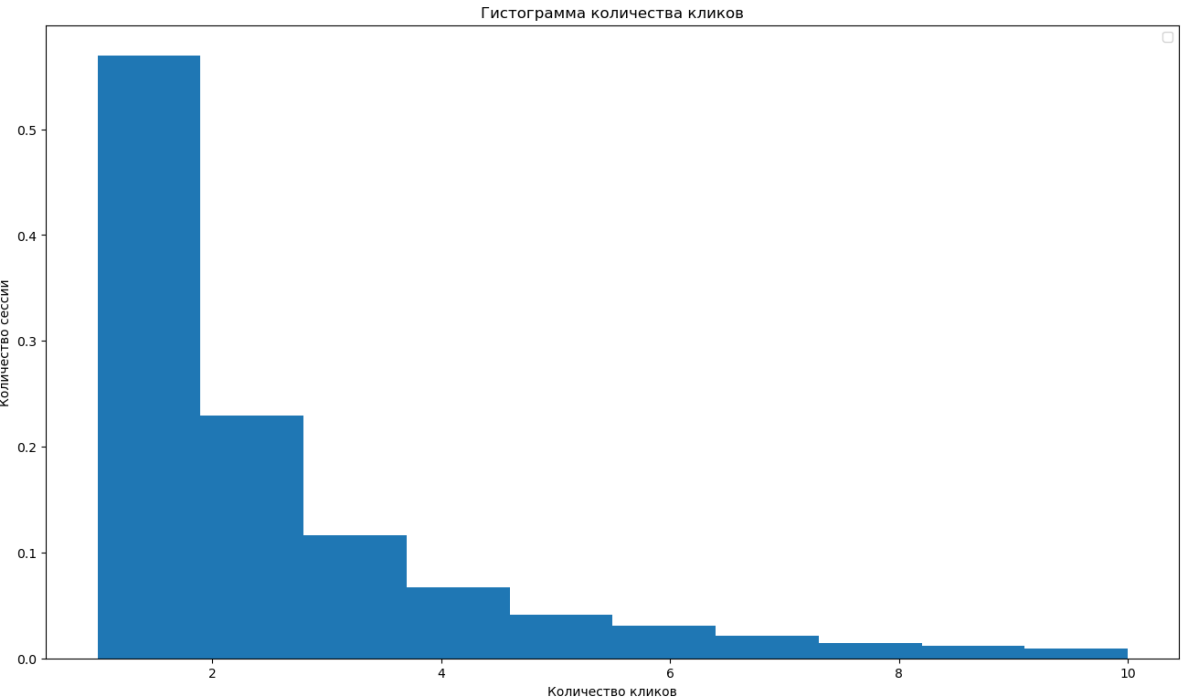
    data['in_group'].value_counts().plot(kind='bar')
    plt.title('Гистограмма количества людей в каждой группе')

    return data
```

```
In [39]: no_exp_data = prepare_data('ab_no_exp_data.csv')
```

No artists with labels found to put in legend. Note that artists whose labels start with an underscore are ignored when legend() is called with no argument.

```
id clicks in_group
0 577919324980 4 False
1 583907240721 2 False
2 561491948916 2 True
3 584748748548 1 False
4 539721118331 1 True
100 percentile: 3054.0
95 percentile: 10.0
85 percentile: 5.0
Среднее в контрольной группе 2.251023229338899
Среднее в тестовой группе 2.2431538927764616
```



# Нулевая гипотеза

Выдвинем нулевую гипотезу  $H_0$  о том, что среднее количество кликов в двух группах равны. И альтернативную  $H_1$ , что они отличаются. Для проверки будем использовать Bootstrap.

```
In [ ]: !pip install bootstrapped
```

```
In [40]: import bootstrapped.bootstrap as bs
import bootstrapped.stats_functions as bs_stats
import bootstrapped.compare_functions as bs_compare
```

10000 итераций, размер каждой новой подвыборки равен размеру исходной выборки.

```
In [41]: def bootstrap(data):
    # Делим данные на тестовую и контрольную группу
    test = data[data['in_group']][['clicks']].values
    ctrl = data[data['in_group'] == False][['clicks']].values # == False логи

    # Запускаем Bootstrap и сохраняем результаты каждой итерации
    boot_results = bs.bootstrap_ab(test=test, ctrl=ctrl, stat_func=bs_stats.
                                   return_distribution=True)

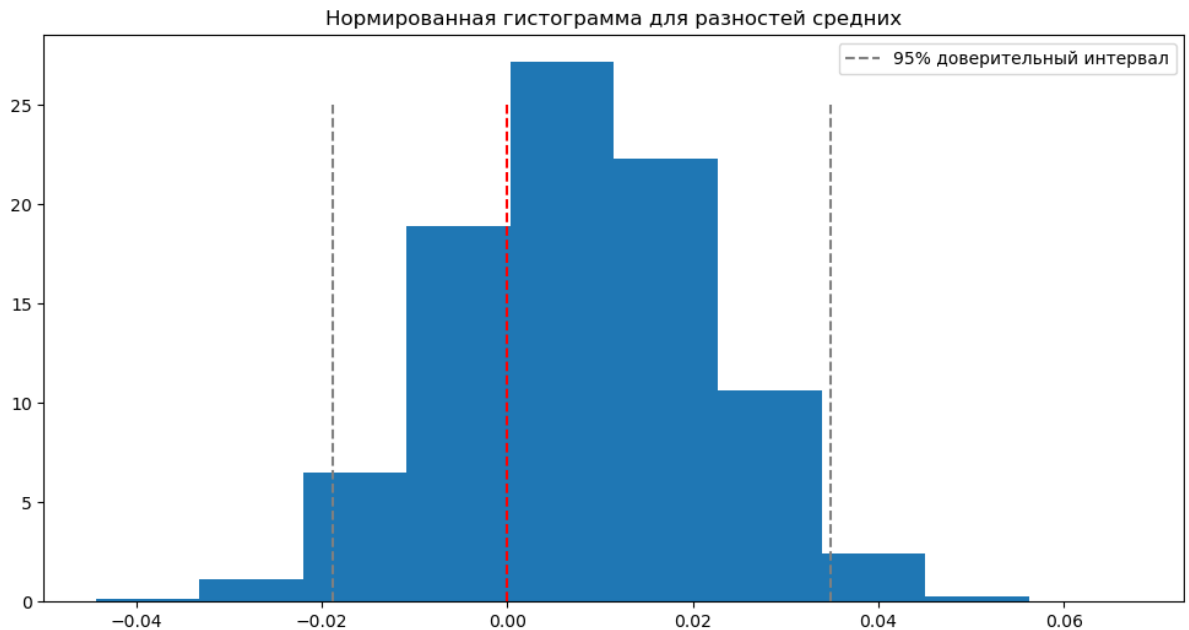
    print(f'Header результатов bootstrap: {boot_results[:10]}')

    borders = np.percentile(boot_results, [2.5, 97.5])
    print('Уровень значимости: 0.05')
    print(f'Границы доверительных интервалов, определенных как 2.5 и 97.5 перцентиль с

    plt.figure(figsize=(12, 6))
    plt.title('Нормированная гистограмма для разностей средних')
    plt.hist(boot_results, density=True)
    plt.vlines(0, ymin=0, ymax=25, linestyle='--', colors='red')
    plt.vlines(borders, ymin=0, ymax=25, linestyle='--', colors='gray', label=borders)
    plt.legend()
    plt.show()
```

```
In [42]: bootstrap(no_exp_data)
```

```
Header результатов bootstrap: [ 0.02023563  0.00941301  0.02813931 -0.00614104
-0.01407847  0.00316723
 0.00654561 -0.00314035  0.02415569 -0.00881692]
Уровень значимости: 0.05
Границы доверительных интервалов, определенных как 2.5 и 97.5 перцентиль соответственно
[-0.01877104  0.03484164]
```



## A/B тест

Прделаем те же самые шаги, но на данных, где действительно проводился эксперимент

```
In [43]: exp_data = prepare_data('ab_exp_data.csv')
```

No artists with labels found to put in legend. Note that artists whose labels start with an underscore are ignored when legend() is called with no argument.

	id	clicks	in_group
0	488919586599	2	True
1	465184427021	1	True
2	454318518857	3	False
3	537597970950	2	False
4	576401480975	1	True

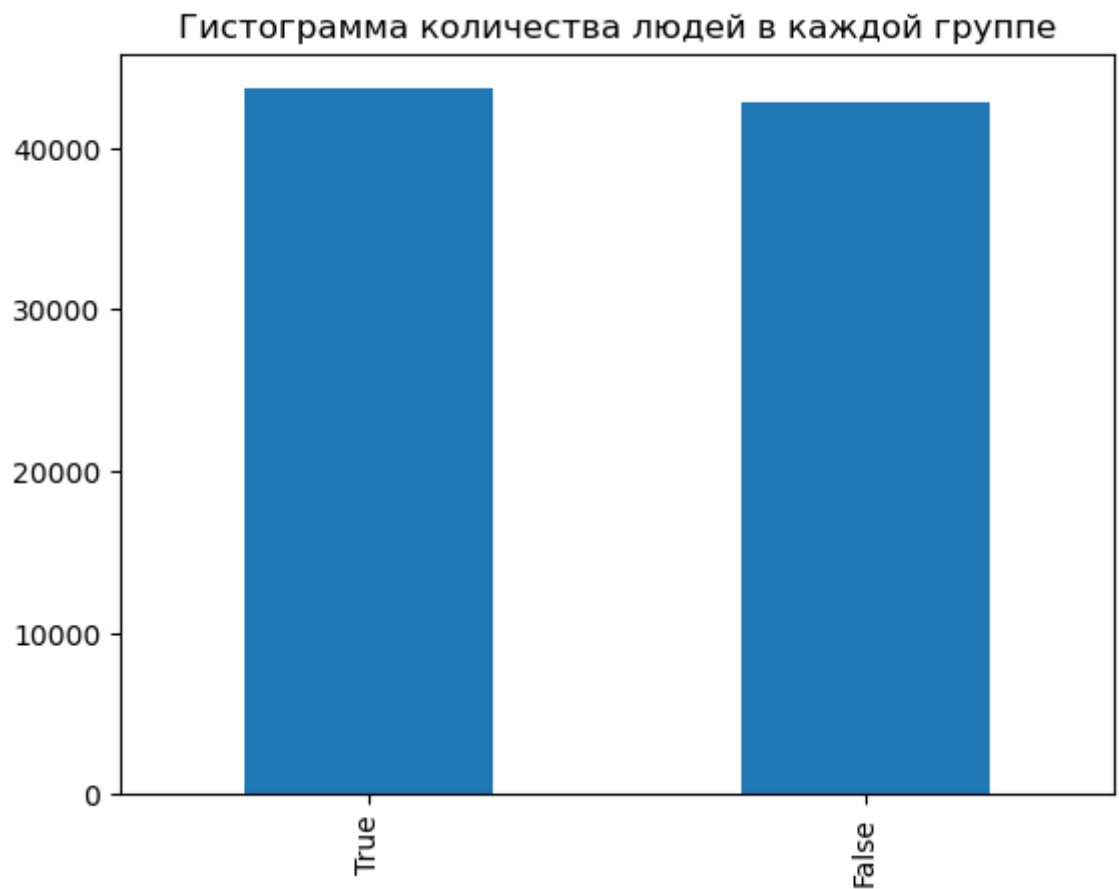
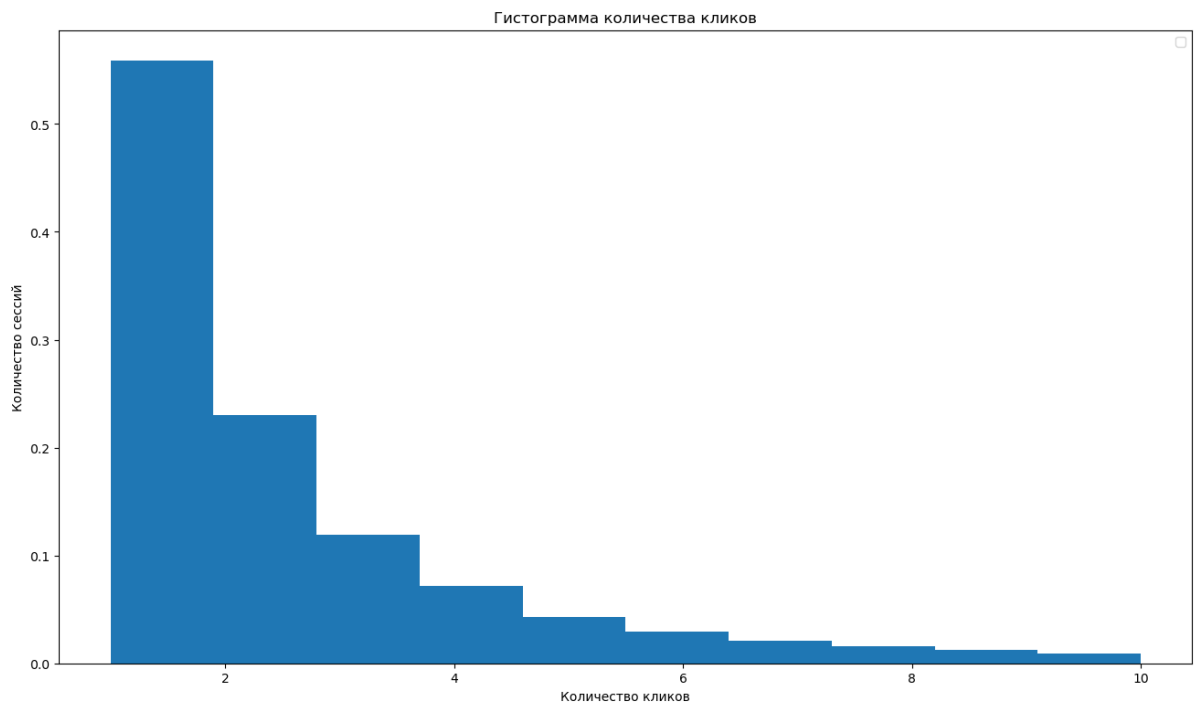
100 percentile: 473.0

95 percentile: 10.0

85 percentile: 5.0

Среднее в контрольной группе 2.3045721361093277

Среднее в тестовой группе 2.251227495908347



In [44]: `bootstrap(exp_data)`

```
Header результатов bootstrap: [0.04879955 0.06534346 0.06634684 0.06463489 0.0
375437 0.04363345
0.04873365 0.05762143 0.04357378 0.04277137]
Уровень значимости: 0.05
Границы доверительных интервалов, определенных как 2.5 и 97.5 перцентиль соответственно
[0.02786903 0.07841645]
```

