[113]:
```
import pandahouse
import scipy.stats as stats
import numpy as np
import pandas as pd
```

[114]:
```
# Подключаемся и извлекаем данные
connection = {
    'host': 'https://clickhouse.lab.karpov.courses',
    'password': 'dpo_python_2020',
    'user': 'student',
    'database': 'simulator'
}

q_all = """
SELECT
    exp_group,
    user_id,
    sum(action = 'like') AS likes,
    sum(action = 'view') AS views
FROM simulator_20250520.feed_actions
WHERE
    toDate(time) BETWEEN '2025-05-02' AND '2025-05-08'
    AND exp_group IN (0, 1, 2, 3)
GROUP BY
    exp_group,
    user_id
HAVING
    views > 0
"""

df = pandahouse.read_clickhouse(q_all, connection=connection)
```

[115]:
```
agg = (df.groupby('exp_group')[['likes', 'views']].sum().reset_index())
agg
```

[115]:

	exp_group	likes	views
0	0	140714	670584
1	1	140339	669543
2	2	132056	659454
3	3	151327	668975

[116]:
```
def get_group_stats(df_agg: pd.DataFrame, group_id: int):
    """возвращает (likes, views, ctr) для заданного exp_group."""
    row = df_agg[df_agg['exp_group'] == group_id]
    likes = int(row['likes'].values[0])
    views = int(row['views'].values[0])
    ctr = likes / views
    return likes, views, ctr
```

[117]:
```
# Агрегированные CTR по группам (сырые CTR)
likes0, views0, ctr0 = get_group_stats(agg, 0)
likes1, views1, ctr1 = get_group_stats(agg, 1)
likes2, views2, ctr2 = get_group_stats(agg, 2)
likes3, views3, ctr3 = get_group_stats(agg, 3)
print(f"Группа 0: likes_0 = {likes0}, views_0 = {views0}, CTR_0 = {ctr0:.6f}")
print(f"Группа 1: likes_1 = {likes1}, views_1 = {views1}, CTR_1 = {ctr1:.6f}")
print(f"Группа 2: likes_2 = {likes2}, views_2 = {views2}, CTR_2 = {ctr2:.6f}")
print(f"Группа 3: likes_3 = {likes3}, views_3 = {views3}, CTR_3 = {ctr3:.6f}")
```

```
Группа 0: likes_0 = 140714, views_0 = 670584, CTR_0 = 0.209838
Группа 1: likes_1 = 140339, views_1 = 669543, CTR_1 = 0.209604
Группа 2: likes_2 = 132056, views_2 = 659454, CTR_2 = 0.200251
Группа 3: likes_3 = 151327, views_3 = 668975, CTR_3 = 0.226207
```

[118]:
```
CTR_control_03 = ctr0
CTR_control_03
```

[118]:
```
0.20983799195924746
```

[119]:
```
df_lin_03 = df.copy()
df_lin_03['linearized_likes'] = df_lin_03['likes'] - CTR_control_03 * df_lin_03['views']
```

[120]:
```
lin0_03 = df_lin_03.loc[df_lin_03['exp_group'] == 0, 'linearized_likes']
lin3_03 = df_lin_03.loc[df_lin_03['exp_group'] == 3, 'linearized_likes']
```

[121]:
```
print(f"Размер выборок: |lin0| = {lin0_03.shape[0]}, |lin3| = {lin3_03.shape[0]}")
print(f"Среднее lin0 = {lin0_03.mean():.6f}, std lin0 = {lin0_03.std(ddof=1):.6f}")
print(f"Среднее lin3 = {lin3_03.mean():.6f}, std lin3 = {lin3_03.std(ddof=1):.6f}")
```

```
Размер выборок: |lin0| = 9920, |lin3| = 10002
Среднее lin0 = 0.000000, std lin0 = 4.799474
Среднее lin3 = 1.094844, std lin3 = 4.747220
```

[122]:
```
t03, p_lin03 = stats.ttest_ind(lin3_03, lin0_03, equal_var=False)
print(f"t-статистика = {t03:.4f}, p-value = {p_lin03:.6e}")
```

```
t-статистика = 16.1862, p-value = 1.491814e-58
```

[ ]:
```
if p_lin03 < 0.05:
    print("p < 0.05: есть статистически значимое отличие линеаризованных лайков. Группы 0 и 3")
else:
    print("p >= 0.05: значимого отличия линеаризованных лайков нет. Группы 0 и 3")
```

```
p < 0.05: есть статистически значимое отличие линеаризованных лайков. Группы 0 и 3
```

Вывод по паре 0 vs 3: Видно ли отличие? Да. Стало ли p-value меньше по сравнению с обычным CTR? Нет.

[124]:
```
CTR_control_12 = ctr1
```

[125]:
```
df_lin_12 = df.copy()
df_lin_12['linearized_likes'] = df_lin_12['likes'] - CTR_control_12 * df_lin_12['views']
```

[126]:
```
lin1_12 = df_lin_12.loc[df_lin_12['exp_group'] == 1, 'linearized_likes']
lin2_12 = df_lin_12.loc[df_lin_12['exp_group'] == 2, 'linearized_likes']
```

[127]:
```
print(f"Размер выборок: |lin1| = {lin1_12.shape[0]}, |lin2| = {lin2_12.shape[0]}")
print(f"Среднее lin1 = {lin1_12.mean():.6f}, std lin1 = {lin1_12.std(ddof=1):.6f}")
print(f"Среднее lin2 = {lin2_12.mean():.6f}, std lin2 = {lin2_12.std(ddof=1):.6f}")
```

```
Размер выборок: |lin1| = 10020, |lin2| = 9877
Среднее lin1 = -0.000000, std lin1 = 4.685238
Среднее lin2 = -0.624512, std lin2 = 9.363371
```

[128]:
```
t12, p_lin12 = stats.ttest_ind(lin2_12, lin1_12, equal_var=False)
print(f"t-статистика = {t12:.4f}, p-value = {p_lin12:.6e}")
```

```
t-статистика = -5.9364, p-value = 2.980506e-09
```

[129]:
```
if p_lin12 < 0.05:
    print("p < 0.05: есть статистически значимое отличие линейрализованных лайков. Группы 1 и 2")
else:
    print("p >= 0.05: значимого отличия линейрализованных лайков нет. Группы 1 и 2")
```

```
p < 0.05: есть статистически значимое отличие линейрализованных лайков. Группы 1 и 2
```

Вывод по паре 1 vs 2: Видно ли отличие? Да. Стало ли p-value меньше по сравнению с обычным CTR? Нет.

[140]:
```
import pandahouse
import pandas as pd
from scipy.stats import ttest_ind

# ============================================================================
# 1. Подключаемся к ClickHouse и вытягиваем raw-данные по группам 1 и 2
# ============================================================================

connection = {
    'host': 'https://clickhouse.lab.karpov.courses',
    'password': 'dpo_python_2020',
    'user': 'student',
    'database': 'simulator'
}

q_12 = """
SELECT
    exp_group,
    user_id,
    sum(action = 'like')  AS likes,
    sum(action = 'view')  AS views
FROM simulator_20250520.feed_actions
WHERE
    toDate(time) BETWEEN '2025-05-02' AND '2025-05-08'
    AND exp_group IN (1, 2)
GROUP BY
    exp_group,
    user_id
HAVING
    views > 0
"""

df_12 = pandahouse.read_clickhouse(q_12, connection=connection)

# ============================================================================
# 2. Считаем per-user CTR = likes / views
# ============================================================================

df_12['ctr'] = df_12['likes'] / df_12['views']

# ============================================================================
# 3. Отдельно выбираем два массива CTR по группам 1 и 2
# ============================================================================

ctr1 = df_12.loc[df_12['exp_group'] == 1, 'ctr']
ctr2 = df_12.loc[df_12['exp_group'] == 2, 'ctr']

# ============================================================================
# 4. Выполняем Welch's t-test для двух несвязанных выборок
# ============================================================================

t_stat, p_value = ttest_ind(ctr2, ctr1, equal_var=False)

print(f"t-статистика (CTR группа 2 vs CTR группа 1) = {t_stat:.4f}")
print(f"p-value                                     = {p_value:.6e}")
```

```
t-статистика (CTR группа 2 vs CTR группа 1) = -6.4051
p-value                                     = 6.853733e-01
```