

```
B [2]: import pandas as pd

df = pd.read_csv ( ' ab_test_data.csv ' )

df.head ( )
```

Out[2]:

	ID пользователя	метка времени	тестовая_группа	конверсия
0	7f6833e6-1141-4f20-b4b2-f1e31019b1fd	2023-07-04 04:40:55.848109	a	0
1	e6a6e960-d3f3-4074-a516-ba1e609b211e	2023-07-06 00:26:45.486187	b	0
2	4d3fbfa5-6847-410a-bac2-477f01d5f400	2023-07-10 20:24:33.639464	b	0
3	361457d9-a044-48f7-981c-d67dc3861679	2023-07-20 07:04:49.957013	b	0
4	285cd63d-7d03-427f-a062-1fa2dd2e77d6	2023-07-19 23:27:50.116680	b	0

```
B [9]: # Количество участников в группе A
count_A = df [ df [ 'test_group' ] == 'a' ] . shape [ 0 ]
# Количество участников в группе B
count_B = df [ df [ 'test_group' ] == 'b' ] . форма [ 0 ]

print ( "Користувачи в групі А:" , count_A )
print ( "Користувачи в групі В:" , count_B )
```

Користувачи в групі А: 10013  
Користувачи в групі В: 9985

```
B [8]: # Количество конверсий в группе A
conv_A = df [ df [ 'test_group' ] == 'a' ] [ 'conversion' ] . sum ( )
# Количество конверсий в группе B
conv_B = df [ df [ 'test_group' ] == 'b' ] [ 'conversion' ] . сума ( )

print ( "Конверсий в групі А:" , conv_A )
print ( "Конверсий в групі В:" , conv_B )
```

Конверсий в групі А: 611  
Конверсий в групі В: 889

```
B [11]: # Ровные конверсии в группе A на отметках
cr_A = conv_A / count_A * 100
# Ровные конверсии в группе B на отметках
cr_B = conv_B / count_B * 100

print ( "CR у групы А (%):" , round ( cr_A , 2 ))
print ( "CR у групы В (%):" , round ( cr_B , 2 ))
```

CR у групы А (%): 6.1  
CR у групы В (%): 8.9

```
B [12]: # Перетворимо колонку в формате datetime
df [ 'timestamp' ] = pd . to_datetime ( df [ 'метка времени' ])

```

```
B [13]: # Дата початку теста
start_date = df [ 'timestamp' ] . min ()
print ( "Данные початку теста:" , start_date.date ( ) )

```

Дата початку теста: 03.07.2023

```
B [14]: # Дата окончания теста
end_date = df [ 'timestamp' ] . max ()
print ( "Дата кінця теста:" , end_date.date ( ) )

```

Дата проведения теста: 25 июля 2023 г.

```
B [15]: # Тривалитет теста в днях
period_days = ( end_date - start_date ) . дней + 1
печать ( "Тривалість теста (дні):" , продолжительность_дней )

```

Тривалитет теста (день): 22

```
B [23]: # --- Z-тест для A/B-теста ---

# Зведена пропорция
p_pool = ( conv_A + conv_B ) / ( count_A + count_B )

# Стандартная поимлка пропорций
импорта math SE = math . sqrt ( p_pool * ( 1 - p_pool ) * ( 1 / count_A + 1 / count_B ) )

# Z-статистика
Z = ( cr_B - cr_A ) / SE

# P -значение для альтернативной гипотезы B > A из scipy.stats
import normal p_value = stats . norm . cdf ( Z )

# --- Выраженные результаты ---
print ( "Статистический критерий:Z-тест для двух пропорциональностей" )
print ( "cr_A:" , round ( cr_A , 4 ) )
print ( "cr_B:" , round ( cr_B , 4 ) )
print ( "Z-статистика:" , round ( Z , 3 ) )
print ( "p-value:" , round ( p_value , 5 ) )

# --- Перестройка нулевой гипотезы ---
альфа = 0.05
if p_value < альфа :
    print ( "Новая гипотеза отображается: конверсия в группе B статистически больше по группе A." )
else :
    print ( "Новая гипотеза не отображается: отрицательное доказательство, что конверсия в группе B статистически больше по группе A." )

```

Статистический критерий:Z-тест для двух пропорциональностей

cr\_A: 6.1021

cr\_B: 8.9034

Z-статистика: 751,968

p-значение: 0,0

Новую гипотезу видно: конверсия в группе Б статистически больше по группе А.

```
B [24]: import matplotlib.pyplot as plt
import numpy as np
from scipy import stats

```

```

# Даные
groups = [ 'A' , 'B' ]
конверсия = [ cr_A , cr_B ] # CR у відсотках
n_users = [ count_A , count_B ]

# Обчислення 95% довірчих інтервалів
conf_intervals = []
for cr, n in zip(conversion, n_users):
    p = cr / 100
    se = np.sqrt(p * (1 - p) / n)
    lower = (p - 1.96 * se) * 100
    upper = (p + 1.96 * se) * 100
    conf_intervals.append((lower, upper))

error_lower = [cr - ci[0] for cr, ci in zip(conversion, conf_intervals)]
error_upper = [ci[1] - cr for cr, ci in zip(conversion, conf_intervals)]
error = [error_lower, error_upper]

# Побудова графіку
fig, ax = plt.subplots(figsize=(6,4))
bars = ax.bar(groups, conversion, yerr=error, capsize=10, color=['skyblue', 'lightgreen'])

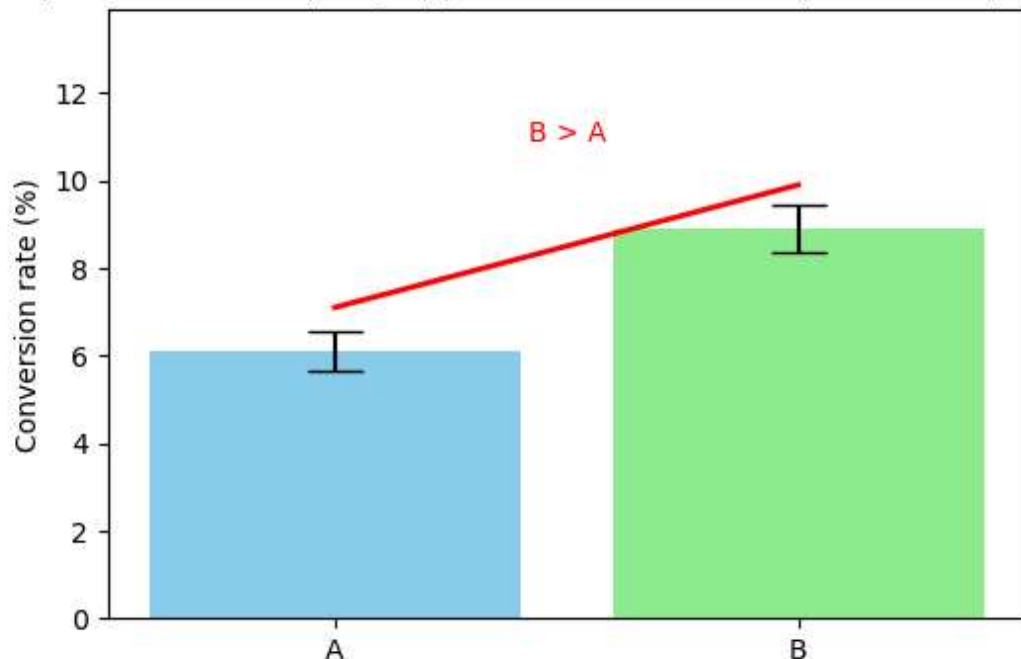
# Додаємо підкреслення різниці між групами
ax.plot([0,1], [conversion[0]+1, conversion[1]+1], color='red', linewidth=2) #
ax.text(0.5, max(conversion)+2, "B > A", color='red', ha='center')

ax.set_ylabel('Conversion rate (%)')
ax.set_title('Порівняння конверсії у групах А і В з 95% довірчими інтервалами')
ax.set_ylim(0, max(conversion)+5)

plt.show()

```

Порівняння конверсії у групах А і В з 95% довірчими інтервалами



## Висновки щодо результатів А/Б-тесту

### Мета тесту

Нульова гіпотеза: пропозиція тижневої підписки \$4,99 зі знижкою 50% не збільшує конверсію серед усіх встановлень додатку.

Альтернативна гіпотеза: конверсія у групі В (з пропозицією) більша, ніж у групі А (контрольна).

### Результати аналізу

Група	Користувачів	Конверсій	Conversion rate (%)
A	10013	611	6.1
B	9985	889	8.9

- Z-статистика: 751.968
- P-value: 0.0 ( $p < 0.05$ )
- Критерій: Z-тест для двох пропорцій

### Інтерпретація статистики

- P-value = 0.0 < 0.05 → нульову гіпотезу відхиляємо.
- Конверсія у групі В статистично значно вища, ніж у групі А.
- Побудовані 95% довірчі інтервали для CR показують, що нижня межа групи В все одно вище за конверсію групи А.

### Висновок та рекомендації

- Пропозиція тижневої підписки \$4,99 зі знижкою 50% покращує конверсію.
- Рекомендується впровадити таку пропозицію для всіх користувачів.
- Результати тесту коректні, оскільки вибірки достатньо великі, а аналіз проведено з урахуванням 95% довірчих інтервалів.

```
In [31]: import matplotlib.pyplot as plt

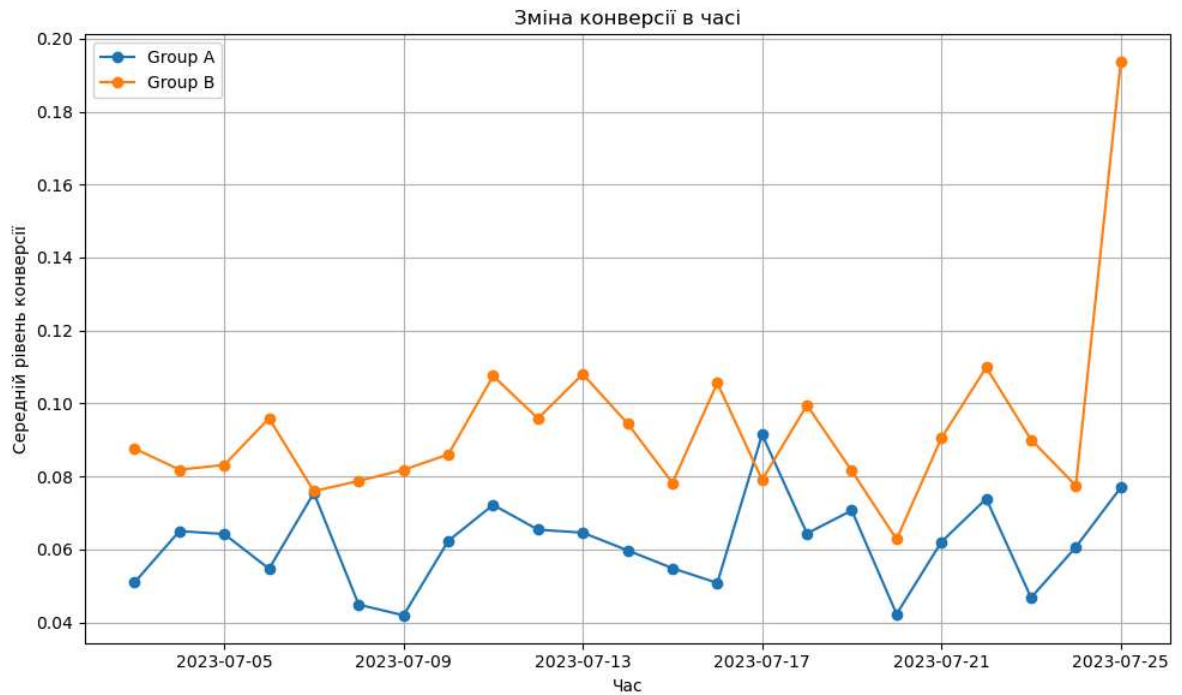
# 1. Переконаємось, що індекс – datetime (у тебе це вже так, але безпечно переві
df.index = pd.to_datetime(df.index)

# 2. Групуємо по часу (наприклад, по днях) і тестовій групі
conversion_time = (
    df
    .groupby([pd.Grouper(freq='D'), 'test_group'])['conversion']
    .mean()
    .reset_index()
)

# 3. Будуємо графік
plt.figure(figsize=(10, 6))

for group in conversion_time['test_group'].unique():
    subset = conversion_time[conversion_time['test_group'] == group]
    plt.plot(
        subset['timestamp'],
        subset['conversion'],
        marker='o',
```

```
    label=f'Group {group.upper()}'  
    )  
  
plt.title('Зміна конверсії в часі')  
plt.xlabel('Час')  
plt.ylabel('Середній рівень конверсії')  
plt.legend()  
plt.grid(True)  
plt.tight_layout()  
plt.show()
```



In [ ]: