Лабораторна 2

«WEB-аналітика»

ФБ-31мп Синчук Владислав

Завдання: На основі даних з [Google Analytics Demo](https://analytics.google.com/analytics/web/?utm_source=demoaccount&utm_medium=demoaccount&utm_campaign=demoaccount#/p213025502) account вибрати ТРИ різні часові ряди і на їх основі шляхом застосування методів визначення аномалій визначити аномалії в поведінці системи.

Я буду використовувати демо акаунт з даними з Google Merchandise Store

(shop.merch.google)

analytics.google.com/analytics/web/

Щоб визначити аномалії у поведінці системи на основі даних з Google

Analytics Demo account, ми можемо використовувати такі часові ряди:

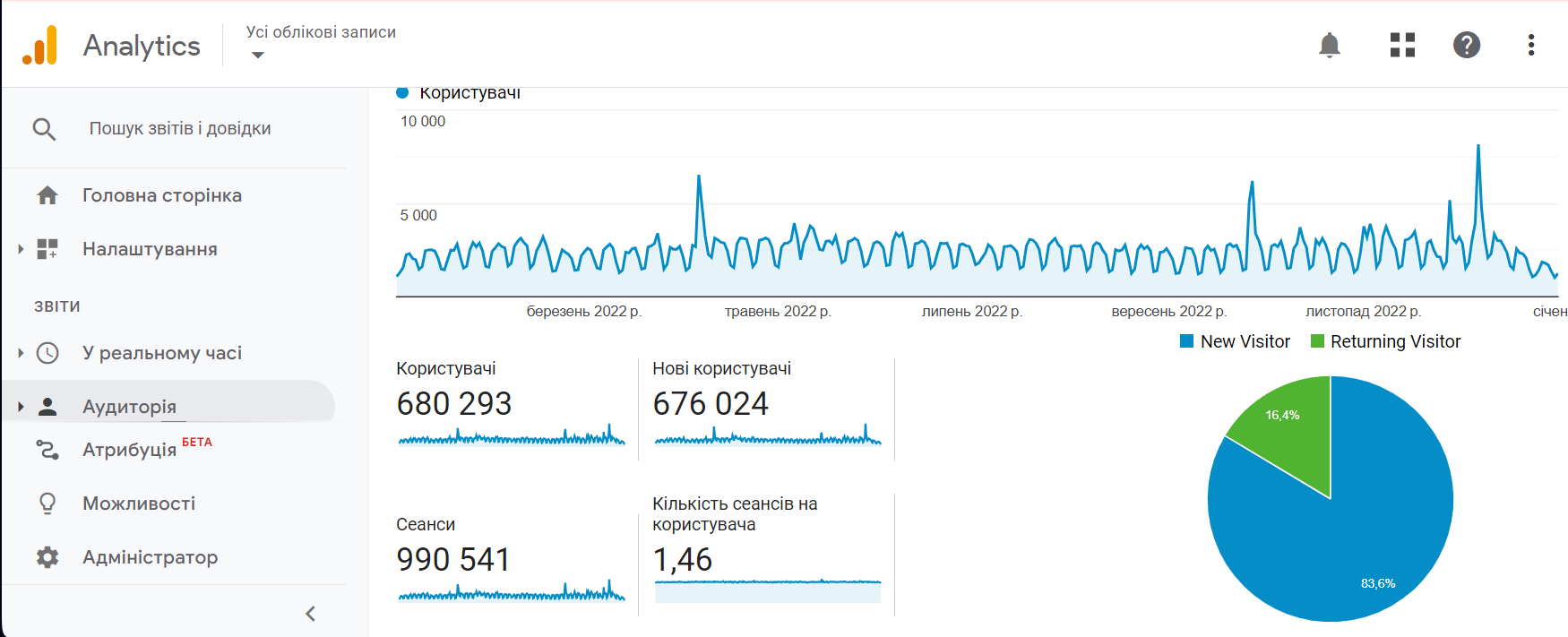
• Сесії (Sessions)

• Користувачі (Users)

• Показник відмов (Bounce Rate)

Дані взято за 2022 рік з кроком в один день





Прочитаємо отримані файли:

users\_path = '/content/drive/MyDrive/web-analytics-synchuk/users.csv'

sessions\_path = '/content/drive/MyDrive/web-analytics-synchuk/sessions.csv'

bounce\_rate\_path = '/content/drive/MyDrive/web-analytics-synchuk/bounce\_rate.csv'

users = pd.read\_csv(users\_path)

sessions = pd.read\_csv(sessions\_path)

bounce\_rate = pd.read\_csv(bounce\_rate\_path)

import re

# Function to extract date and number from a string

def extract\_date\_and\_number(s: str) -> tuple:

    # Define regular expressions to match the date and the number

    date\_pattern = r'(\d{2}\.\d{2}\.\d{2})'

    number\_pattern = r'"(\d+),(\d+)"'

    # Search for the date pattern in the string

    date\_match = re.search(date\_pattern, s)

    if not date\_match:

        print("No matching date found in the input string")

    # Search for the number pattern in the string

    number\_match = re.search(number\_pattern, s)

    if not number\_match:

        print("No matching number found in the input string")

    # Extract the date

    date\_str = date\_match.group(1)

    # Extract the matched number and replace the comma with a dot

    number\_str = number\_match.group(1) + '.' + number\_match.group(2)

    number\_float = float(number\_str)

    return date\_str, number\_float

s = '01.01.22,"4,07"'

print(extract\_date\_and\_number(s))

new\_bounce\_rate = pd.DataFrame()

new\_bounce\_rate[['Індекс дня', 'Сторінок за сеанс']] = bounce\_rate['Індекс дня'][:-1].apply(lambda x: pd.Series(extract\_date\_and\_number(x)))

# for row in bounce\_rate['Індекс дня'][:-1]:

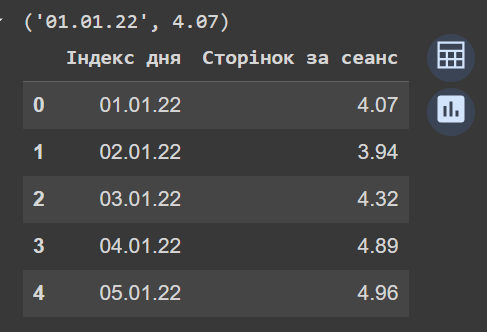
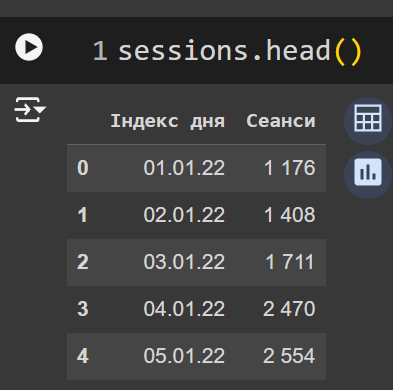
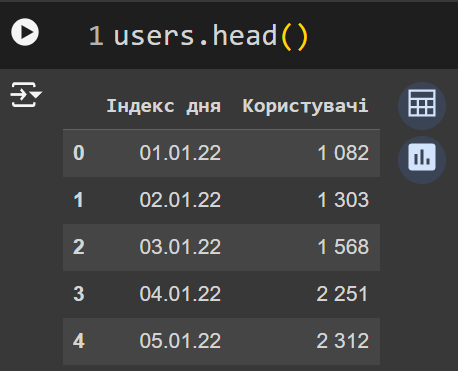
#     print(extract\_date\_and\_number(row))

#     # print(row)

bounce\_rate = copy.deepcopy(new\_bounce\_rate)

bounce\_rate.head()

Прочитані файли:

Конвертуємо рядки в дату:

users['Індекс дня'] = pd.to\_datetime(users['Індекс дня'], format='%d.%m.%y')

sessions['Індекс дня'] = pd.to\_datetime(sessions['Індекс дня'], format='%d.%m.%y')

bounce\_rate['Індекс дня'] = pd.to\_datetime(bounce\_rate['Індекс дня'], format='%d.%m.%y')

