# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

## Кафедра прикладних інформаційних систем

**Звіт до лабораторної роботи №1**

# з курсу «Big Data»

*Студента 4 курсу*

*Групи ПП-41*

*«Комп'ютерні науки»*

*ОП«Прикладне програмування»*

*Постоюка Максима*

## Київ – 2023

# Тема

## Агрегація, обробка пропусків та візуалізація даних пакетами Python

# Мета

Метою лабораторної роботи є отримання практичних навичок у роботі з raw data, використовуючи пакети jupyter, pandas, seaborn.

# Завдання

### Завантажити файли з даними у папку проекту з посилання: <https://github.com/oreillymedia/doing_data_science>

### Створіть нову змінну age\_group, яка агрегує користувачів як <18, 18–24, 25–34, 35–44, 45–54, 55–64 та 65+.

### Зафіксуйте на діаграмі кількість показів та показник переходів (CTR = #clicks/#inmpressions) для цих шести вікових категорій.

### Вивчіть дані та проведіть візуальні та кількісні порівняння між сегментами користувачів/демографічними групами (наприклад, чоловіки старше 18 років у порівнянні з жінками старше 18 років або авторизовані та неавторизовані користувачі).

### Створіть метрики/вимірювання/статистику, які підсумовують дані. Приклади можливих метрик включають CTR, квантил, середнє значення, медіану, дисперсію та максимальне значення. Ці показники потрібно розрахувати за різними сегментами користувачів. Подумайте про елементи, які важливо відстежувати з часом - що стискає дані, але, як і раніше, захоплює поведінку користувача.

### Результати статистичного дослідження подати у вигляді результуючого ДатаФрейма (одного), дивлячись на який можна зрозуміти і порівнювати дані за віковими підкатегоріями.

### Опишіть та інтерпретуйте будь-які закономірності, які знайдете.

### Завантажити файл ipynb з виконаними завданнями на git в окрему папку з відповідною назвою лабораторної роботи

# Виконання завдання

iimport pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import matplotlib as mpl

import seaborn as sns

mpl.rcParams["font.size"] = 18

def divide\_two\_cols(df\_sub):

    return df\_sub['Clicks'].sum() / float(df\_sub['Impressions'].sum())

df = pd.concat(

    map(pd.read\_csv, ['dds\_ch2\_nyt/nyt1.csv', 'dds\_ch2\_nyt/nyt2.csv', 'dds\_ch2\_nyt/nyt3.csv', 'dds\_ch2\_nyt/nyt4.csv', 'dds\_ch2\_nyt/nyt5.csv'

     , 'dds\_ch2\_nyt/nyt6.csv', 'dds\_ch2\_nyt/nyt7.csv', 'dds\_ch2\_nyt/nyt8.csv', 'dds\_ch2\_nyt/nyt9.csv', 'dds\_ch2\_nyt/nyt10.csv', 'dds\_ch2\_nyt/nyt11.csv'

      , 'dds\_ch2\_nyt/nyt12.csv', 'dds\_ch2\_nyt/nyt13.csv', 'dds\_ch2\_nyt/nyt14.csv', 'dds\_ch2\_nyt/nyt15.csv', 'dds\_ch2\_nyt/nyt16.csv', 'dds\_ch2\_nyt/nyt17.csv'

      , 'dds\_ch2\_nyt/nyt18.csv'   , 'dds\_ch2\_nyt/nyt19.csv'   , 'dds\_ch2\_nyt/nyt20.csv'   , 'dds\_ch2\_nyt/nyt21.csv'   , 'dds\_ch2\_nyt/nyt22.csv'   , 'dds\_ch2\_nyt/nyt23.csv'

      , 'dds\_ch2\_nyt/nyt24.csv'   , 'dds\_ch2\_nyt/nyt25.csv'   , 'dds\_ch2\_nyt/nyt26.csv'   , 'dds\_ch2\_nyt/nyt27.csv'   , 'dds\_ch2\_nyt/nyt28.csv'   , 'dds\_ch2\_nyt/nyt29.csv'

      , 'dds\_ch2\_nyt/nyt30.csv'   , 'dds\_ch2\_nyt/nyt31.csv']), ignore\_index=True)

# create a list of our conditions

conditions = [

    (df['Age'] < 18),

    (df['Age'] >= 18) & (df['Age'] <= 24),

    (df['Age'] >= 25) & (df['Age'] <= 34),

    (df['Age'] >= 35) & (df['Age'] <= 44),

    (df['Age'] >= 45) & (df['Age'] <= 54),

    (df['Age'] >= 55) & (df['Age'] <= 64),

    (df['Age'] >= 65)

    ]

# create a list of the values we want to assign for each condition

values = ['18-', '18-24', '25-34', '35-44', '45-54', '55-64', '65+']

# create a new column and use np.select to assign values to it using our lists as arguments

df['Age group'] = np.select(conditions, values)

df.replace({'Gender': {0 :'Female'}}, inplace=True)

df.replace({'Gender': {1 :'Male'}}, inplace=True)

plt.show()

df2 = df.groupby('Age group').apply(divide\_two\_cols).reset\_index(name='CTR')

plotCRT = sns.barplot(x='Age group', y= 'CTR', data=df2, order=['18-', '18-24', '25-34', '35-44', '45-54', '55-64', '65+'])

plotCRT.set(title = "Users' CTR by age groups")

plt.show()

signedInPercentageDF = df

signedInPercentageDF['Percentage'] = (df['Signed\_In'] /

                  df['Signed\_In'].sum()) \* 100

df3 = signedInPercentageDF.groupby('Age group')['Percentage'].sum().reset\_index()

plotSignedIn = sns.barplot(x='Age group', y= 'Percentage', data=df3, order=['18-', '18-24', '25-34', '35-44', '45-54', '55-64', '65+'])

plotSignedIn.set(title = "Signed In users by age groups")

plt.show()

df4 = signedInPercentageDF.groupby(['Age group', 'Gender'])['Percentage'].sum().reset\_index()

df4.drop(df4[df4['Age group'] == "18-"].index, inplace=True)

df5 = df4.groupby('Gender')['Percentage'].sum().reset\_index()

plotSignedIn = sns.barplot(x='Gender', y= 'Percentage', data=df5)

plotSignedIn.set(title = "Signed In users who is older than 18 years old by gender")

plt.show()

dfImpressionsSum = df.groupby('Age group')['Impressions'].sum().reset\_index(name = 'Total Impressions')

plotImpressionsSum =  sns.barplot(x='Age group', y= 'Total Impressions', data=dfImpressionsSum)

plotImpressionsSum.set(title = 'Total impressions by age groups')

plt.show()

dfImpressionsMean = df.groupby('Age group')['Impressions'].mean().reset\_index(name = 'Impressions mean')

plotImpressionsMean =  sns.barplot(x='Age group', y= 'Impressions mean', data=dfImpressionsMean)

plotImpressionsMean.set(title = 'Impressions\' mean by age groups')

plt.show()

fig, ax = plt.subplots()

sns.histplot(

    data=df, x='Impressions', hue='Age group', multiple='dodge', bins = 22).set(title="Impressions by age groups histogram")

plt.show()

dfImpressionsMedian = df.groupby('Age group')['Impressions'].median().reset\_index(name = 'Impressions Median')

plotImpressionsMedian =  sns.scatterplot(x='Age group', y= 'Impressions Median', data=dfImpressionsMedian)

plotImpressionsMedian.set(title = 'Impressions median by age groups')

plt.show()

dImpressionsMax = df.groupby('Age group')['Impressions'].max().reset\_index(name = 'Impressions Max')

plotImpressionsMax =  sns.scatterplot(x='Age group', y= 'Impressions Max', data=dImpressionsMax)

plotImpressionsMax.set(title = 'Impressions max by age groups')

plt.show()

dfImpressionsDispersion = df.groupby('Age group')['Impressions'].std().reset\_index(name = 'Impressions Dispersion')

plotImpressionsDispersion =  sns.scatterplot(x='Age group', y= 'Impressions Dispersion', data=dfImpressionsDispersion)

plotImpressionsDispersion.set(title = 'Impressions dispersion by age groups')

plt.show()

dfClicksSum = df.groupby('Age group')['Clicks'].sum().reset\_index(name = 'Total Clicks')

plotClicksSum =  sns.barplot(x='Age group', y= 'Total Clicks', data=dfClicksSum)

plotClicksSum.set(title = 'Total clicks by age groups')

plt.show()

dfClicksMean = df.groupby('Age group')['Clicks'].mean().reset\_index(name = 'Clicks Mean')

plotClicksMean =  sns.barplot(x='Age group', y= 'Clicks Mean', data=dfClicksMean)

plotClicksMean.set(title = 'Clicks\' mean by age groups')

plt.show()

fig, ax = plt.subplots()

sns.histplot(

    data=df, x='Clicks', hue='Age group', multiple='dodge', bins = 22).set(title="Clicks by age groups histogram")

plt.show()

dfClicksMedian = df.groupby('Age group')['Clicks'].median().reset\_index(name = 'Clicks Median')

plotClicksMedian =  sns.scatterplot(x='Age group', y= 'Clicks Median', data=dfClicksMedian)

plotClicksMedian.set(title = 'Clicks median by age groups')

plt.show()

dfClicksMax = df.groupby('Age group')['Clicks'].max().reset\_index(name = 'Clicks Max')

plotClicksMax =  sns.scatterplot(x='Age group', y= 'Clicks Max', data=dfClicksMax)

plotClicksMax.set(title = 'Clicks max by age groups')

plt.show()

dfClicksDispersion = df.groupby('Age group')['Clicks'].std().reset\_index(name = 'Clicks Dispersion')

plotClicksDispersion =  sns.scatterplot(x='Age group', y= 'Clicks Dispersion', data=dfClicksDispersion)

plotClicksDispersion.set(title = 'Clicks dispersion by age groups')

plt.show()

# Статистика CTR

# 

# З графіка можемо зробити висновок, що найбільший CTR у користувачів, яким більше 65 років, та у користувачів, яким менше 18 років. Найменший у користувачів, яким від 18 до 64 років.

# Статистика зареєстрованих користувачів

# 

# З графіка можемо зробити висновок, щонайбільше зареєстрованих користувачів у віковій категорії 35-44(близько 24%), а найменше у віковій категорії <18(близько 4%).

# 

# З графіка можемо зробити висновок, чоловіків серед зареєстрованих користувачів близько на 3% більше, ніж жінок.

# Підсумок статистики реакцій

# 

# Найбільша кількість реакцій у користувачів, яким менше 18(близько 30 мільйонів), а найменша у користувачів, яким більше 65(близько 5 мільйонів).

# 

# Середнє значення у всіх вікових груп однакове і дорівнює 5.

# 

# Бачимо, що найбільшу кількість реакцій робит вікова група 18-, а найбільша кількість реакцій це 4 та 6. Найменша кількість реакцій - 0, 1 та усі, що більше 12.

# 

# На графіку бачимо, що медіана реакцій для всіх вікових груп - 5.

# 

# З графіку робимо висновок, що максимальна кількість реакцій(21) серед вікової групи 0-18.

# 

# Найбільша дисперсія реакцій у вікової групи 0 - 18 = 2.240, а найменша у вікової групи 65+ = 2.233.

# Підсумок статистики кліків

# 

# Найбільша кількість кліків у користувачів, яким менше 18 років(близько 850000), а найменша у користувачів, яким від 18 до 24 років(близько 60000).

# 

# Найбільші середні значення кліків у вікових груп 18- та 65+(0.14 та 0.15 відповідно), а найменше у вікових груп від 25 до 54 років(0.05).

# Бачимо, що найбільша кількість кліків - 0, найбільше робить їх вікова група 18-. Найменша кількість кліків - 3, вікова група 18- робить їх найбільше.

# Збільшені зображення на кліки 2 та 3 відповідно:

# 

# Продовжуємо аналіз кліків.

# 

# Медіана кліків однакова у всіх вікових груп - 0.

# 

# Найбільша кількість кліків у вікової групи 0-18 = 6, а найменша у вікових груп 25-34 та 45-54 = 3.

# 

# Найбільша дисперсія кліків у вікових груп 0-18 та 65+ - близько 0.39 , а найменша у вікових груп 25-34, 35-44 та 45-54 - близько -0.23.

**Висновок**

Отримав практичні навички у роботі з raw data, використовуючи пакети jupyter, pandas, seaborn.