## Лабораторна робота № 1.

## Агрегація, обробка пропусків та візуалізація даних пакетами Python

Виконав студент групи ПП-41:

Шостак В. О.

Викладач: Білий Р.О.

**Завдання**

### Завдання до виконання

### 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.

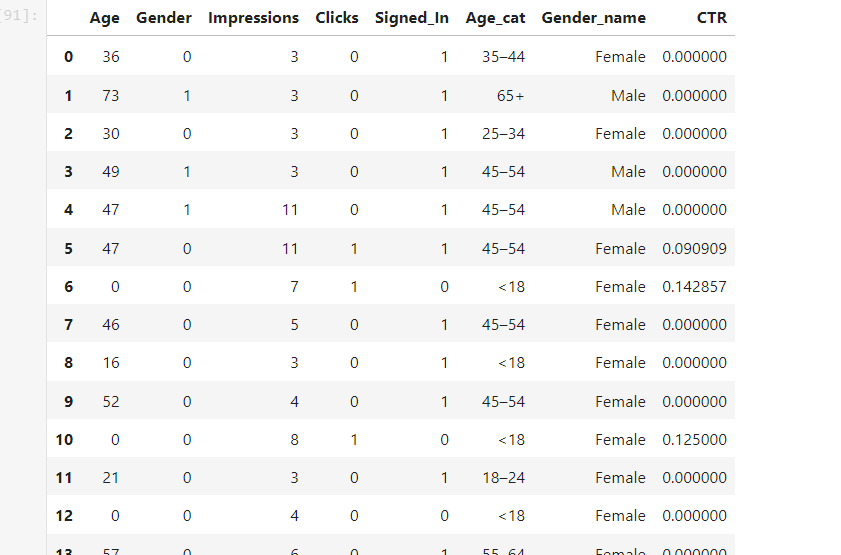
### Результати статистичного дослідження подати у вигляді результуючого ДатаФрейма (одного), дивлячись на який можна зрозуміти і порівнювати дані за віковими підкатегоріями.

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

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

**Хід роботи**

1. Добавляємо вікові групи як <18, 18–24, 25–34, 35–44, 45–54, 55–64 та 65+ в колонку Age\_cat та рахуэмо CTR (clicks/impressions).



bins = [-float('inf'), 18, 25, 35, 45, 55, 65, float('inf')]

agg\_y = ['<18', '18–24', '25–34', '35–44', '45–54', '55–64', '65+']

df.loc[df['Age'] <18, 'Age\_cat'] = '<18'

df.loc[(df['Age'] >=18) & (df['Age'] <=24), 'Age\_cat'] = '18–24'

df.loc[(df['Age'] >=25) & (df['Age'] <=34), 'Age\_cat'] = '25–34'

df.loc[(df['Age'] >=35) & (df['Age'] <=44), 'Age\_cat'] = '35–44'

df.loc[(df['Age'] >=45) & (df['Age'] <=54), 'Age\_cat'] = '45–54'

df.loc[(df['Age'] >=55) & (df['Age'] <=64), 'Age\_cat'] = '55–64'

df.loc[df['Age'] >=65, 'Age\_cat'] = '65+'

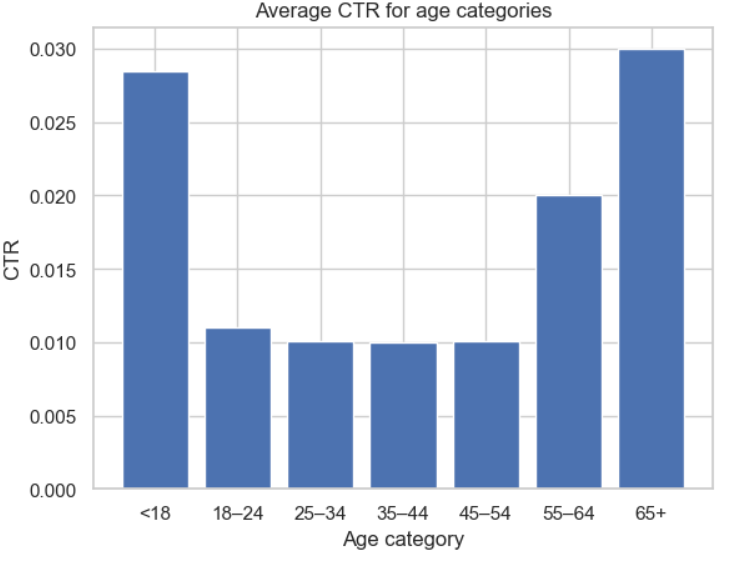
df.loc[df['Gender'] == 0, 'Gender\_name'] = 'Female'

df.loc[df['Gender'] == 1, 'Gender\_name'] = 'Male'

df['CTR']= df['Clicks'] / df['Impressions']

df['CTR'].fillna(-999.0, inplace=True)

2. Середній CTR для вікових категорій. Бачимо, що найбільші CTR для вікових груп 65+ та <18.



filtered\_data1 = df[df['CTR'] != -999].groupby('Age\_cat')['CTR'].mean().reset\_index()

age\_categories\_order = ['<18', '18–24', '25–34', '35–44', '45–54', '55–64', '65+']

filtered\_data1['Age\_cat'] = pd.Categorical(filtered\_data1['Age\_cat'], categories=age\_categories\_order, ordered=True)

sorted\_age1 = filtered\_data1.sort\_values(by='Age\_cat')

sorted\_age1

sns.set\_style("whitegrid")

plt.bar(sorted\_age1['Age\_cat'], sorted\_age1['CTR'])

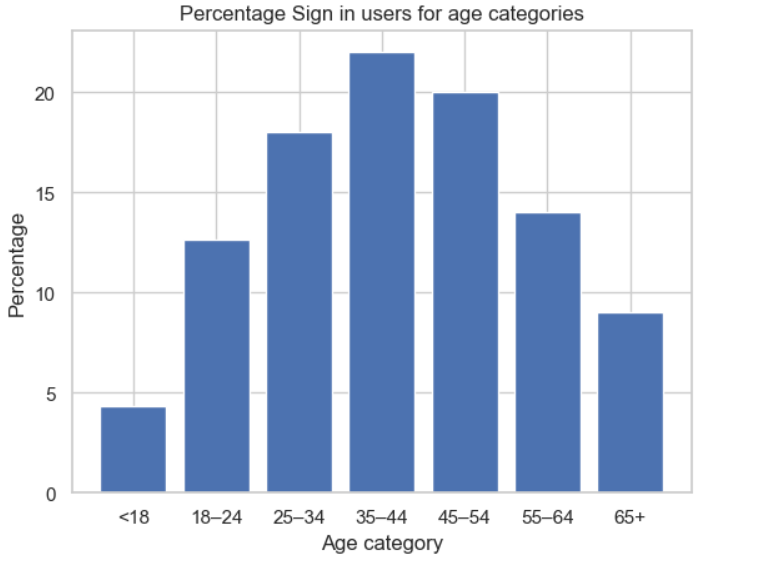
plt.xlabel('Age category')

plt.ylabel('CTR')

plt.title('Average CTR for age categories')

plt.show()

3. Відсоток користувачів, які зайшли до облікового запису для вікових категорій. Найбільший показник для вікової групи 35-44.



filtered\_data1 = df.groupby('Age\_cat')['Signed\_In'].sum().reset\_index()

all\_val\_num = filtered\_data1['Signed\_In'].sum()

filtered\_data2 = filtered\_data1

filtered\_data2['Signed\_In'] = (filtered\_data1['Signed\_In'] / all\_val\_num \* 100)

age\_categories\_order = ['<18', '18–24', '25–34', '35–44', '45–54', '55–64', '65+']

filtered\_data2['Age\_cat'] = pd.Categorical(filtered\_data2['Age\_cat'], categories=age\_categories\_order, ordered=True)

sorted\_age1 = filtered\_data1.sort\_values(by='Age\_cat')

sns.set\_style("whitegrid")

plt.bar(sorted\_age1['Age\_cat'], sorted\_age1['Signed\_In'])

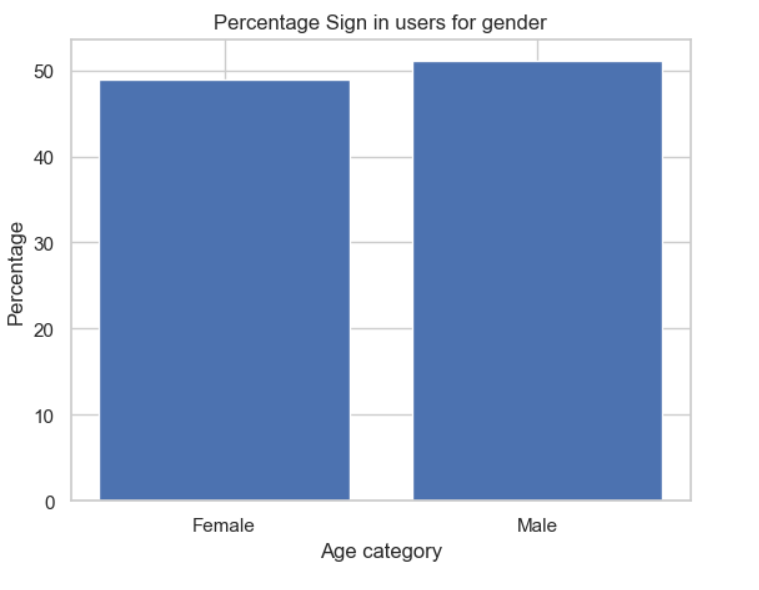
plt.xlabel('Age category')

plt.ylabel('Percentage')

plt.title('Percentage Sign in users for age categories')

plt.show()

4. Відсоток користувачів, які зайшли до облікового запису для статі людини. Відсотох для обох гендерів майже однаковий, що показує, що немає різниці для зареєстрованих користувачів для різних гендерів.



filtered\_data1 = df[df['Age'] >= 18].groupby('Gender\_name')['Signed\_In'].sum().reset\_index()

all\_val\_num = filtered\_data1['Signed\_In'].sum()

filtered\_data2 = filtered\_data1

filtered\_data2['Signed\_In'] = (filtered\_data1['Signed\_In'] / all\_val\_num \* 100)

sns.set\_style("whitegrid")

plt.bar(filtered\_data2['Gender\_name'], filtered\_data2['Signed\_In'])

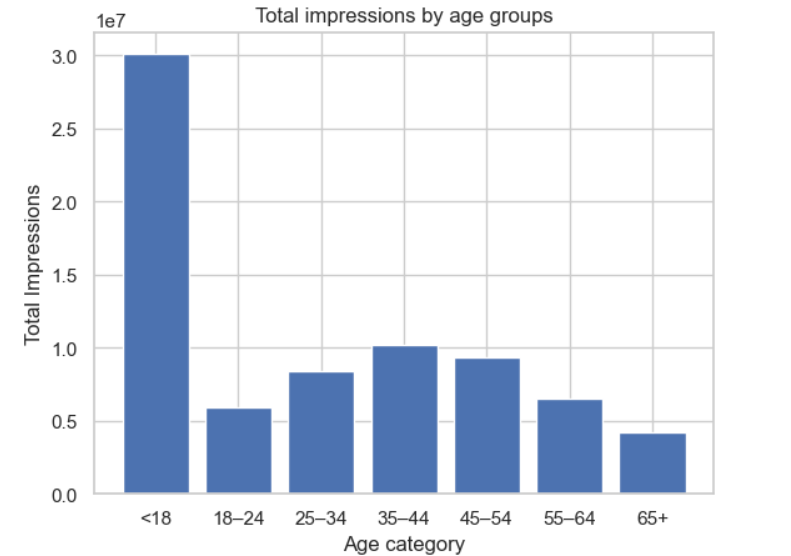
plt.xlabel('Age category')

plt.ylabel('Percentage')

plt.title('Percentage Sign in users for gender')

plt.show()

5. Загальна кількість показів за віковими групами. Як бачимо найбільша загальна кількість для вікової групи <18 , що показує, що для молодого покоління видається більша кількість показів.



filtered\_data1 = df.groupby('Age\_cat')['Impressions'].sum().reset\_index()

age\_categories\_order = ['<18', '18–24', '25–34', '35–44', '45–54', '55–64', '65+']

filtered\_data1['Age\_cat'] = pd.Categorical(filtered\_data1['Age\_cat'], categories=age\_categories\_order, ordered=True)

sorted\_age1 = filtered\_data1.sort\_values(by='Age\_cat')

sns.set\_style("whitegrid")

plt.bar(sorted\_age1['Age\_cat'], sorted\_age1['Impressions'])

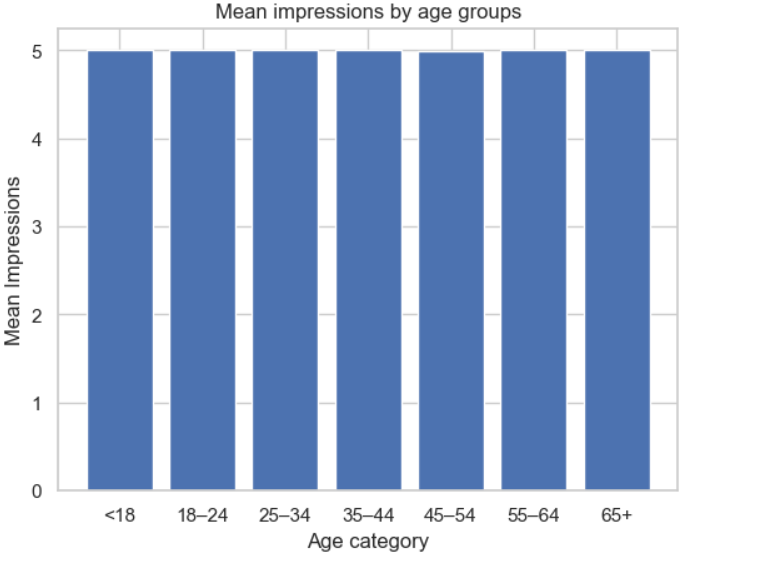
plt.xlabel('Age category')

plt.ylabel('Total Impressions')

plt.title('Total impressions by age groups')

plt.show()

6. Середні покази за віковими групами. Як бачимо для всіх вікових груп однакові середні покази.



filtered\_data1 = df.groupby('Age\_cat')['Impressions'].mean().reset\_index()

age\_categories\_order = ['<18', '18–24', '25–34', '35–44', '45–54', '55–64', '65+']

filtered\_data1['Age\_cat'] = pd.Categorical(filtered\_data1['Age\_cat'], categories=age\_categories\_order, ordered=True)

sorted\_age1 = filtered\_data1.sort\_values(by='Age\_cat')

sns.set\_style("whitegrid")

plt.bar(sorted\_age1['Age\_cat'], sorted\_age1['Impressions'])

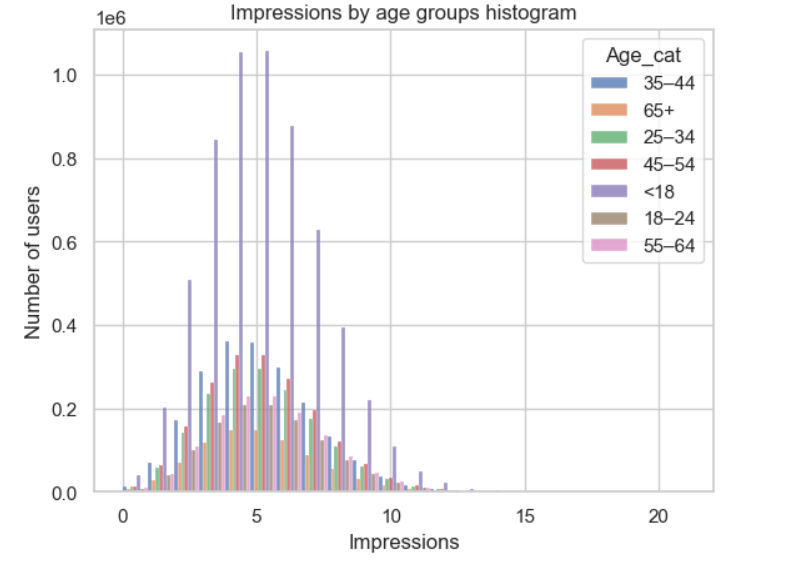
plt.xlabel('Age category')

plt.ylabel('Mean Impressions')

plt.title('Mean impressions by age groups')

plt.show()

7. Гістограма показів за віковими групами. На гістограмі відображена кількість користувачів для кожної вікової групи в порівнянні з кількістю показів, що показує, що вікова група <18 має найбільшу кількість користувачів та в значеннях показів близько п’яти.



fig, ax = plt.subplots()

sns.histplot(

data=df, x='Impressions', hue='Age\_cat', multiple='dodge', bins = 22).set(title="Impressions by age groups histogram", ylabel = 'Number of users')

plt.show()

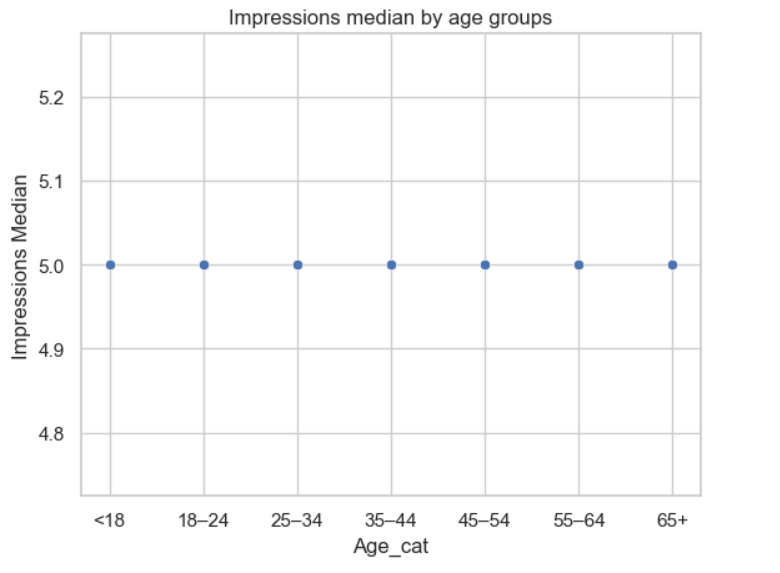
8. Медіана показів за віковими групами.

Медіана - це статистичний показник, який розташовує дані у впорядкованому ряді та розділяє їх на дві рівні частини. Якщо маємо непарну кількість значень, то медіана буде середнім значенням. Якщо кількість значень парна, то медіана визначається середнім арифметичним двох центральних значень.

Застосовуючи медіану до показників за віковими групами, ми розташовуємо людей у впорядкованому порядку за віком, а потім знаходимо медіану показів для кожної групи.

Медіана не чутлива до великих відхилень або екстремальних значень, що може бути важливо при розгляді вікових груп, де може бути значна різниця в віці між невеликою кількістю людей.

Як бачимо для всіх вікових груп медіана показів однакова.



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

age\_categories\_order = ['<18', '18–24', '25–34', '35–44', '45–54', '55–64', '65+']

dfImpressionsMedian['Age\_cat'] = pd.Categorical(dfImpressionsMedian['Age\_cat'], categories=age\_categories\_order, ordered=True)

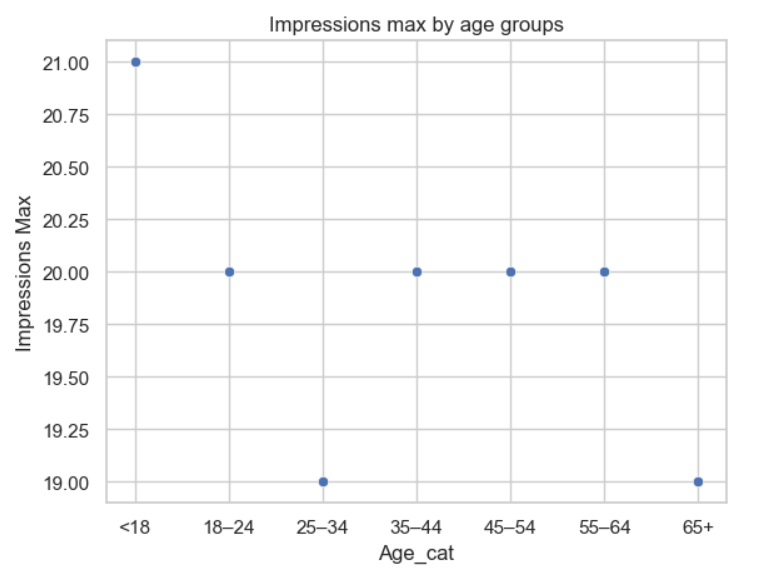
sorted\_age1 = dfImpressionsMedian.sort\_values(by='Age\_cat')

plotImpressionsMedian = sns.scatterplot(x='Age\_cat', y= 'Impressions Median', data=sorted\_age1)

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

plt.show()

9. Максимальне значення показів за віковими групами. Як бачимо максимальне значення належить віковій групі <18 .



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

age\_categories\_order = ['<18', '18–24', '25–34', '35–44', '45–54', '55–64', '65+']

dImpressionsMax['Age\_cat'] = pd.Categorical(dImpressionsMax['Age\_cat'], categories=age\_categories\_order, ordered=True)

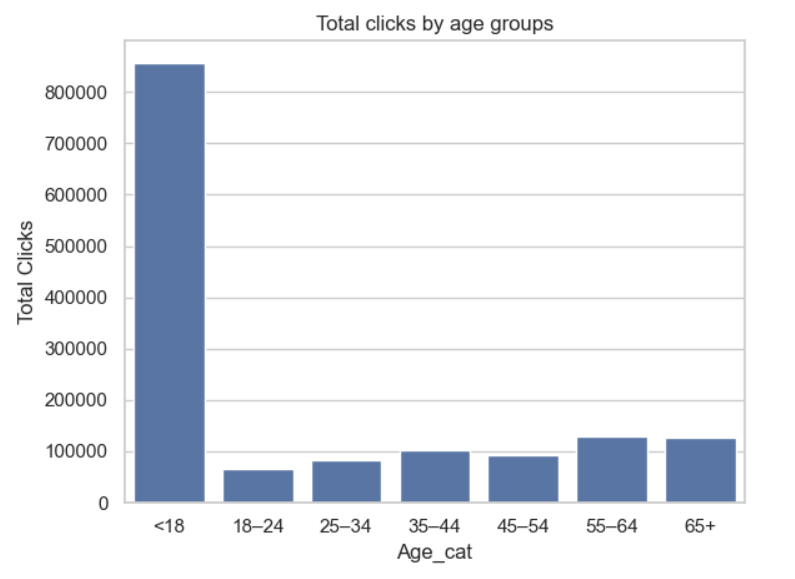
sorted\_age1 = dImpressionsMax.sort\_values(by='Age\_cat')

plotImpressionsMax = sns.scatterplot(x='Age\_cat', y= 'Impressions Max', data=sorted\_age1)

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

plt.show()

10. Загальна кількість кліків за віковими групами. Як бачимо найбільша загальна кількість кліків для вікової групи <18 .



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

age\_categories\_order = ['<18', '18–24', '25–34', '35–44', '45–54', '55–64', '65+']

dfClicksSum['Age\_cat'] = pd.Categorical(dfClicksSum['Age\_cat'], categories=age\_categories\_order, ordered=True)

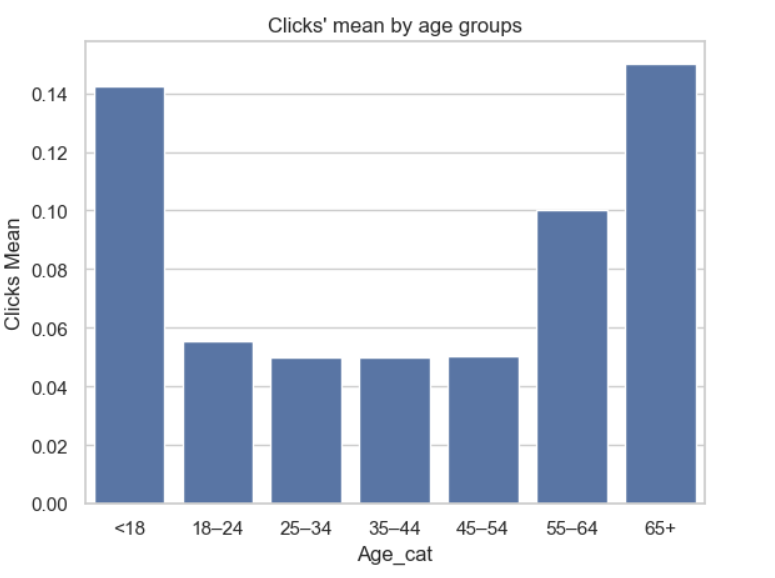
sorted\_age1 = dfClicksSum.sort\_values(by='Age\_cat')

plotClicksSum = sns.barplot(x='Age\_cat', y= 'Total Clicks', data=sorted\_age1)

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

plt.show()

11. Середня кількість кліків за віковими групами. Як бачимо для вікових груп <18 та 65+ найбільша середня кількість кліків.



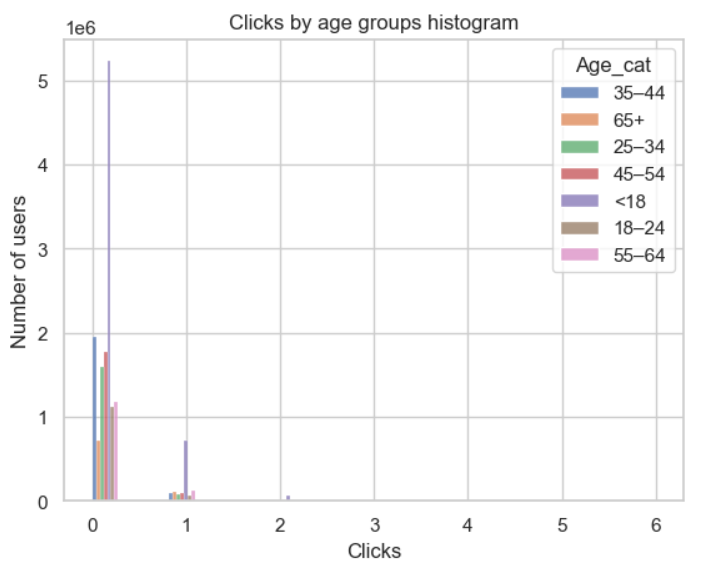
dfClicksMean = df.groupby('Age\_cat')['Clicks'].mean().reset\_index(name = 'Clicks Mean')

plotClicksMean = sns.barplot(x='Age\_cat', y= 'Clicks Mean', data=sort\_age\_group(dfClicksMean))

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

plt.show()

12. Гістограма кліків за віковими групами. Гістограма кліків за віковими групами може бути корисним інструментом для візуалізації та розуміння розподілу кліків. Як бачимо для вікової групи <18 найбільша кількість коритувачів та її значення знахоиться в межах від 0 до 1.



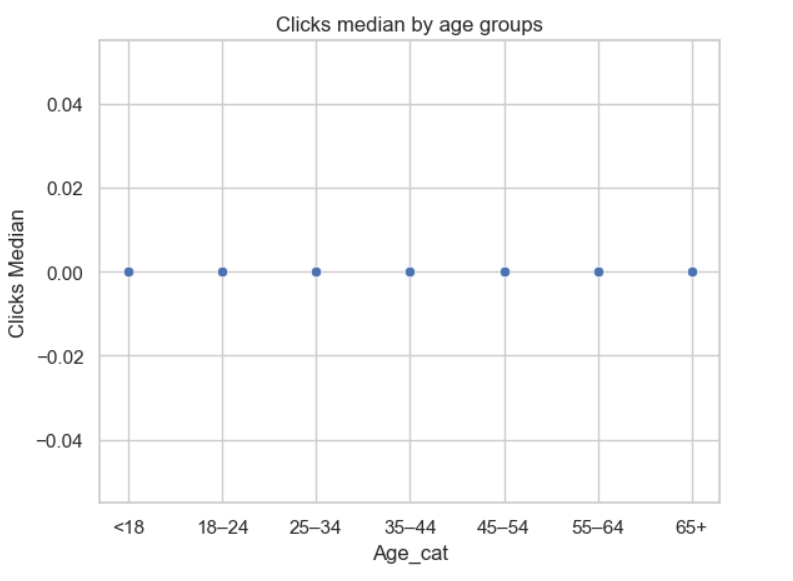
fig, ax = plt.subplots()

sns.histplot(

data=df, x='Clicks', hue='Age\_cat', multiple='dodge', bins = 22).set(title="Clicks by age groups histogram", ylabel = 'Number of users')

plt.show()

13. Медіана кліків за віковими групами. Медіана кліків за віковими групами може бути корисною для низки цілей і аналізу під час оцінювання ефективності контенту, реклами чи інших взаємодій на основі віку користувачів. Як бачимо медіана клідків для всіх вікових груп однакова.



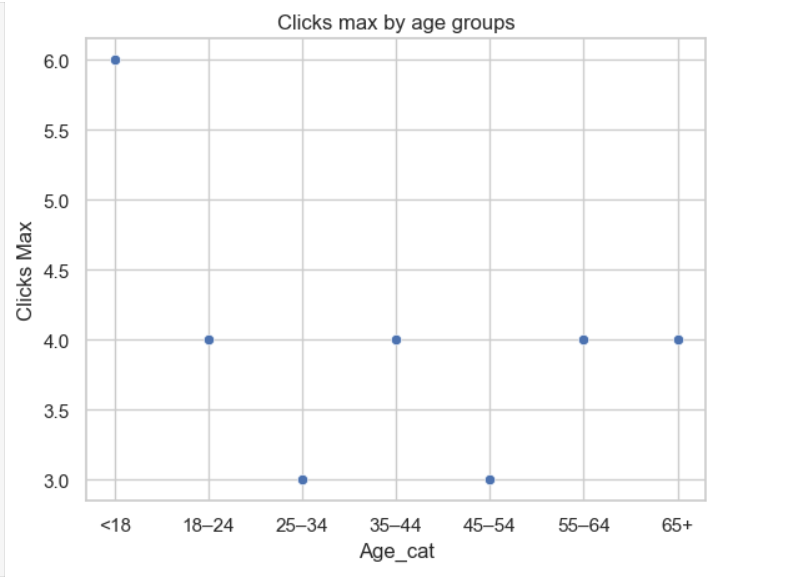
dfClicksMedian = df.groupby('Age\_cat')['Clicks'].median().reset\_index(name = 'Clicks Median')

plotClicksMedian = sns.scatterplot(x='Age\_cat', y= 'Clicks Median', data=sort\_age\_group(dfClicksMedian))

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

plt.show()

14. Максимальна кількість кліків за віковими групами. Як бачимо для вікової групи <18 найбільша максимальна кількість кліків.



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

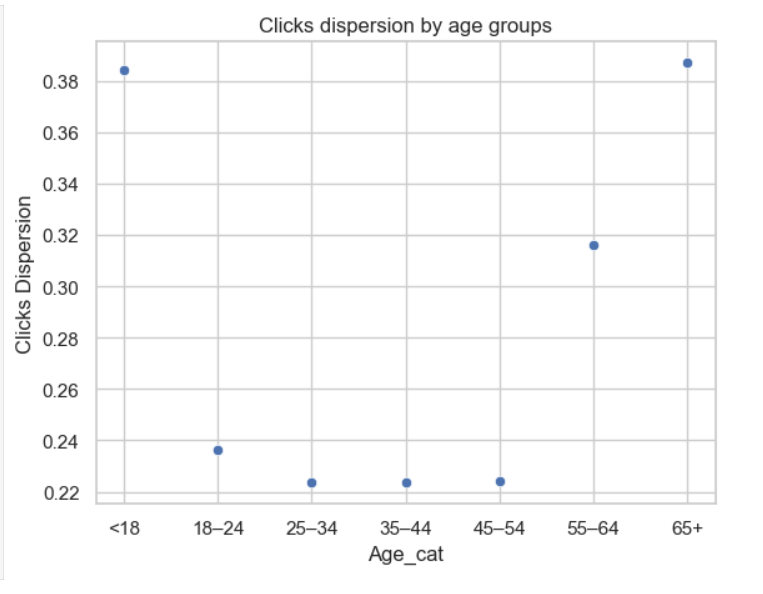
plotClicksMax = sns.scatterplot(x='Age\_cat', y= 'Clicks Max', data=sort\_age\_group(dfClicksMax))

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

plt.show()

15. Дисперсія кліків за віковими групами.

Дисперсія є мірою розкиду або варіабельності значень в масиві даних. Вона може бути корисною для вивчення того, наскільки великі коливання відбуваються в розподілі кліків за віковими групами. Як бачимо найбільші коливання значеннь для вікових груп <18 та 65+.

****

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

plotClicksDispersion = sns.scatterplot(x='Age\_cat', y= 'Clicks Dispersion', data=sort\_age\_group(dfClicksDispersion))

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

plt.show()