

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ  
ТАРАСА ШЕВЧЕНКА  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Звіт до лабораторної роботи №3  
на тему  
**«Аналіз розмірів файлів»**

Студента 3 курсу ФКНК  
групи ТТП-31  
Корнієнка Олександра

Київ-2024

## **Зміст**

<b><i>Вступ .....</i></b>	<b><i>3</i></b>
<b><i>Код програми .....</i></b>	<b><i>4</i></b>
<b><i>Приклад виконання.....</i></b>	<b><i>6</i></b>

## Вступ

Проаналізувати частотну характеристику (частотний розподіл) розміру файлів у файловій системі на вашому комп'ютері (залежність кількості файлів від їх розміру).

Надати звіт:

1) підготовка даних: як саме збирали дані про розподіл для аналізу

\* hint: це можна зробити одним командним рядком у ОС (find / ls + grep + awk + > result.txt)

2) власне аналіз та **візуалізація залежностей**

\* hint: гістограма кількості файлів (Oy) залежно від їх розміру (Ox) - гарна відправна точка, але цього явно недостатньо

3) висновки

\* hint: гарний висновок – "переважна більшість файлів (x%) має розміри у діапазоні від a до b", де x - якомога більше (75-80-85-90%), а [a, b] – якомога вужчий проміжок

Hint: краще брати всю файлову систему (або принаймні системний диск, а не лише 1 каталог).

## Код програми

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.widgets import Slider

file_sizes = pd.read_csv('file_sizes.txt', header=None, names=['size'])

intervals = [0, 1024, 1024*1024, 10*1024*1024, 100*1024*1024, 1024*1024*1024,
float('inf')]
labels = ['<1KB', '1KB-1MB', '1MB-10MB', '10MB-100MB', '100MB-1GB', '>1GB']

file_sizes['size_interval'] = pd.cut(file_sizes['size'], bins=intervals, labels=labels,
right=False)

file_size_counts = file_sizes['size_interval'].value_counts().sort_index()

fig, ax = plt.subplots(figsize=(10, 9))
bar_plot = ax.bar(file_size_counts.index, file_size_counts.values, color='skyblue')
ax.set_title(f'Гістограма кількості файлів за їх розміром (усього {len(file_sizes)} файлів)')
ax.set_xlabel('Розмір файлу (байт)')
ax.set_ylabel('Кількість файлів')
ax.grid(axis='y')
ax.set_xticklabels(labels=file_size_counts.index, rotation=0, ha='center')

for i, v in enumerate(file_size_counts.values):
    ax.text(i, v, str(v), ha='center', va='bottom', rotation=0)

ax_slider = plt.axes([0.15, 0.01, 0.7, 0.03])
slider = Slider(ax_slider, 'Кількість файлів', 25, max(file_size_counts),
valinit=max(file_size_counts), valstep=25)
```

```
def update(val):
    max_val = int(slider.val)
    ax.set_ylim(0, max_val)
    plt.draw()

slider.on_changed(update)

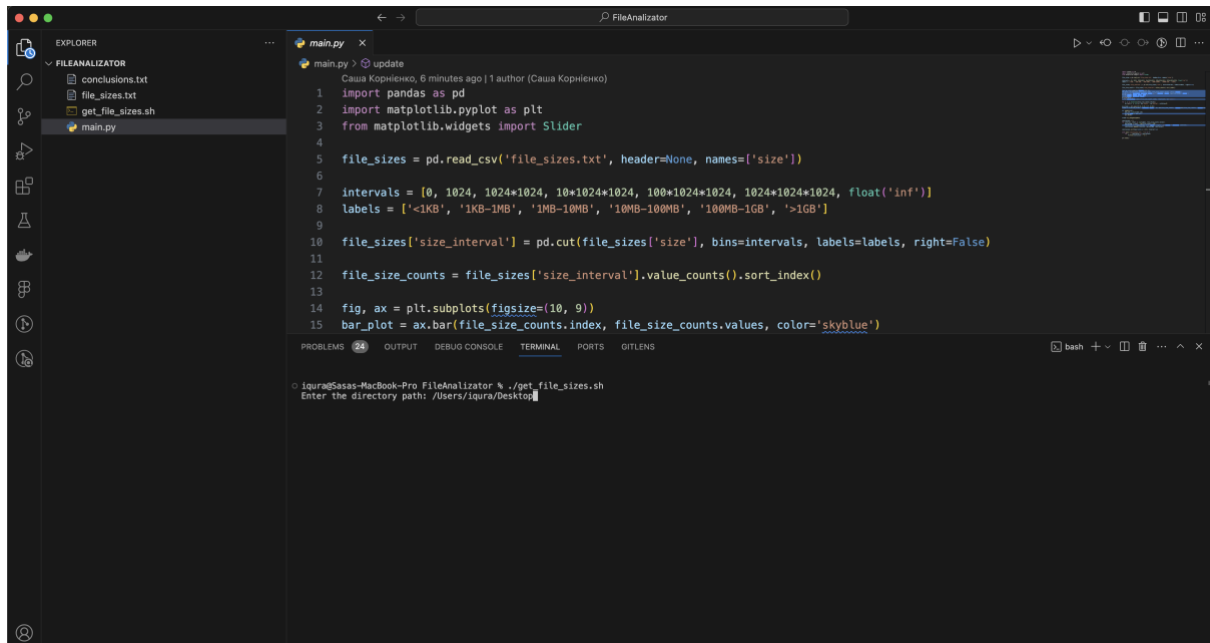
conclusions = []
for interval, count in zip(labels, file_size_counts.values):
    percentage = count / len(file_sizes) * 100
    conclusion = f"({percentage:.2f}%) файлів ({count}) має розміри у діапазоні {interval}."
    conclusions.append((interval, percentage, conclusion))

conclusions.sort(key=lambda x: x[1], reverse=True)

with open('conclusions.txt', 'w') as f:
    for _, _, conclusion in conclusions:
        f.write(conclusion + '\n')

plt.show()
```

# Приклад виконання



The image shows a Visual Studio Code editor window with a project named "FILEANALIZATOR". The Explorer sidebar on the left shows the project structure with files: conclusions.txt, file\_sizes.txt, get\_file\_sizes.sh, and main.py. The main editor displays the code for main.py, which uses pandas and matplotlib to analyze file sizes and create a bar chart. The code includes imports, reading a CSV file, defining intervals and labels, using pd.cut to categorize file sizes, and finally plotting a bar chart with plt.subplots and ax.bar. The bottom of the editor shows a terminal window with the command ./get\_file\_sizes.sh being executed, prompting for a directory path.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 from matplotlib.widgets import Slider
4
5 file_sizes = pd.read_csv('file_sizes.txt', header=None, names=['size'])
6
7 intervals = [0, 1024, 1024*1024, 10*1024*1024, 100*1024*1024, 1024*1024*1024, float('inf')]
8 labels = ['<1KB', '1KB-1MB', '1MB-10MB', '10MB-100MB', '100MB-1GB', '>1GB']
9
10 file_sizes['size_interval'] = pd.cut(file_sizes['size'], bins=intervals, labels=labels, right=False)
11
12 file_size_counts = file_sizes['size_interval'].value_counts().sort_index()
13
14 fig, ax = plt.subplots(figsize=(10, 9))
15 bar_plot = ax.bar(file_size_counts.index, file_size_counts.values, color='skyblue')
```

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

iquradSasas-MacBook-Pro FileAnalyzer % ./get\_file\_sizes.sh  
Enter the directory path: /Users/iqura/Desktop

