

# DS exam simulation, November 5<sup>th</sup>, 2024

## Download the data

1. Consider the file [sciprogram-ds-05-09-2024-FIRSTNAME-LASTNAME-ID.zip](#) and extract it on your desktop.
2. Rename [sciprogram-ds-05-09-2024-FIRSTNAME-LASTNAME-ID](#) folder:

Replace **FIRSTNAME**, **LASTNAME**, and **ID** with your first name, last name and student id number. Failure to comply with these instructions will result in the loss of 1 point on your grade.

like [sciprogram-ds-05-09-2024-alessandro-romanel-432432](#)

From now on, you will be editing the files in that folder.

3. Edit the files following the instructions.
4. At the end of the exam, **compress** the folder in a zip file

[sciprogram-ds-05-09-2024-alessandro-romanel-432432.zip](#)

and submit it. This is what will be evaluated. Please, include in the zip archive all the files required to execute your implementations!

**NOTE:** You can only use the data structures and packages provided in the exam script files. **Importing other Python packages IS NOT allowed** unless explicitly stated in the exam instructions. Using Python collections or other libraries will impact your final grade. Still, **IT IS ALLOWED** to use **built-in Python operators** as we have done during the practical classes (max, min, len, reversed, list comprehensions, etc).

## Exercise 1 [FIRST MODULE]

This dataset provides a comprehensive analysis of mobile device usage patterns and user behavior classification. It contains 700 samples of user data, including metrics such as app usage time, screen-on time, battery drain, and data consumption. Each entry is categorized into one of five user behavior classes, ranging from light to extreme usage, allowing for insightful analysis and modeling.

The dataset is stored in the data folder and it looks like:

	User ID	Device Model	Operating System	App Usage Time (min/day)	Screen On Time (hours/day)	Battery Drain (mAh/day)	Number of Apps Installed	Data Usage (MB/day)	Age	Gender	User Behavior Class
0	1	Google Pixel 5	Android	393	6.4	1872	67	1122	40	Male	4
1	2	OnePlus 9	Android	268	4.7	1331	42	944	47	Female	3
2	3	Xiaomi Mi 11	Android	154	4.0	761	32	322	42	Male	2
3	4	Google Pixel 5	Android	239	4.8	1676	56	871	20	Male	3
4	5	iPhone 12	iOS	187	4.3	1367	58	988	31	Female	3

1. load the dataset
2. Define a function named `max_usage` that takes the loaded dataset as input and **prints** the age of the user with the highest app usage time.

for instance:

```
> max_usage(data)
>> 58
```

3. Define a function named `OS_count` that takes three inputs: a dataset and two integers, `min_age` and `max_age`. The function should filter the dataset to include only users whose ages fall within the specified range (`min_age ≤ age ≤ max_age`). It should then **return** a dictionary with two keys, `"Android"` and `"iOS"`, where each key's value represents the number of users within the age range who use that specific operating system.  
NB: you have to set the **default** value of `min_age = 18` and `max_age = 40`

for instance:

```
> OS_count(data,min_age=30,max_age=35)
>> {'Android': 80, 'iOS': 24}
```

4. Write a function named `data_usage_by_age_and_os` that takes the dataset as input and organizes data usage information by both age group and operating system.

First, divide users into three age groups:

Young: ages 18–29

Middle-aged: ages 30–49

Senior: ages 50 and above

For each age group and operating system ("Android" and "iOS"), calculate:

The average daily data usage.

The total number of users in that age group and OS category.

The function should **return** a nested dictionary with the following structure:

```
{
  "Young": {
    "Android": {"average_data_usage": X, "user_count": Y},
    "iOS": {"average_data_usage": A, "user_count": B}
  },
  "Middle-aged": {
    "Android": {"average_data_usage": M, "user_count": N},
    "iOS": {"average_data_usage": O, "user_count": P}
  },
  "Senior": {
    "Android": {"average_data_usage": Q, "user_count": R},
    "iOS": {"average_data_usage": S, "user_count": T}
  }
}
```

5. **Store** the returned dictionary (from exercise 4) in a file named `dict.txt`.
6. Build a plot showing the OS histogram and the Age histogram. The plot should have two subfigures, pay attention to putting the x/y labels and titles. Finally, store the plot as `my_plot.pdf`.

