Dear students,

please read this carefully.

The exam is designed so that you *do not get stuck in any part*.
Your goal is to show us that you can think like a programmer, handle real-world issues with processing data, and break the task into logical pieces.

You will most likely run into trouble. Do not give up! Coding is hard, and you will need to gain a lot of experience, so we suggest approaching this with a cool head. Remember to use Google and **stackoverflow.com**.

It is prohibited to share codes with each other! It is prohibited to use generative AI tools such as Chat GPT.

INSTRUCTIONS:

- Make sure you know your CUNI student number (like in your email 12345678@fsv.cuni.cz).
- **DEADLINE: 19:50, 25th of November 2024**
- Work preferably with jupyter lab or jupyter notebook application.
- Present your work in a jupyter notebook (.ipynb) in a GitHub repository, which you had in through this form: **https://forms.gle/yTCh7Z9qBkmX63Jd8**
    - Only **pushed** commits that fall within the time allowed will be considered.

- There should be no unreasonable delay between the deadline and the submission to the Google form.
- Your code **needs to run from scratch without errors**.
- In total, you can receive **25** points.
- Generally, the topics tested are in lectures and seminars until Pandas II + Matplotlib.

If you show a decent attempt, **you can receive points for partial work!**

Happy coding!

Data analysis – 25 points

We suggest creating a pandas DataFrame with the data. You might want to create a DF with dates on the index, and a column per company.

PART 1 (15pts)

Get the data (0 points):

- Download your custom dataset from https://ies-fsv.s3.eu-central-1.amazonaws.com/studentsSets/XXXXXXX.zip
- where XXXXXXX stands for your CUNI student number.
  - Make sure you copy the link correctly (no spaces, other characters (open it in a text editor first)).
  - If you cannot find the file, let us know!
  - It is a simple zip file, you should be able to open on Windows, Mac, unix easily. There is *no need to use python* for this, you are welcome to do it manually and unzip using your favorite tools.
- Make sure you can access the CSV files within from your Jupyter notebook.
- The name of the file is a ticker of a company, do not loose/rename it, you will need it.
- Make sure that Dates are correctly represented as **Datetime** variables
  In the following analysis, **use the Close time series**, unless specified otherwise.

7x 2pt tasks + 1x 1pt task:

1. Is there a company that has no difference between the Open and Close columns? What does it mean from the financial point of view for the stock (you can get bonus partial points)?
2. What is the highest and lowest price (Close) each company recorded?
3. (1pt task) Calculate *logarithmic* returns from Close. For each company report on its, *min, man, mean, median* of the return distribution.
4. When did each company record the highest gain and highest loss for the day? (logarithmic loss). *Hint: idxmax*
5. What is the average calendar weekly volume for each company? *Hint: check how to resample pandas DF*
6. Which company recorded the highest total return over the whole period?
7. Create a new column `volume_class` based on the volume column into categories (e.g., "Low", "Medium", "High", "Very High") and use quartile thresholds for the classes.
8. A. Plot the log-returns of the companies (ideally in the same plot).
   B. Show the log-return distribution of the companies (ideally in the same plot).

SEE PART 2 on another page

PART 2 (10pts)

Download the dataset about all S&P 500 companies https://ies-fsv.s3.eu-central-1.amazonaws.com/companies/companies_no_subindustry.csv. Note that `pd.read_csv` can directly read public URLs.

5x 2pt tasks:

1. Find out how many companies do not filled-in the date of inclusion (column "included") to S&P 500.
2. Delete the companies with no inclusion date and calculate which company is the oldest/youngest constituent and tell us the average age of a constituent in the sample. If you need to fix anything or make any assumptions, comment on them in the code.
   Hint: *pd.to_datetime (some date column, dayfirst=True, errors='coerce')*
3. Describe the distribution of companies across sectors and create a plot that demonstrates the proportionality of the sectors (i.e. pie plot, or something like this)
4. Parse the "hq" column, extract the state of the hq and describe the distribution of the states
5. Join the dataset with this one: https://ies-fsv.s3.eu-central-1.amazonaws.com/companies/companies_subindustry.csv
   And join the two datasets based on an appropriate key. Report on distribution of subindustries for the "Consumer Discretionary" GICS sector.

SUBMISSION CHECKPOINTS:

- !!! Save your work !!!
- Make sure your notebook (.ipynb) runs from scratch – restart the kernel, import everything and check it works
- Commit and **push** (commits not pushed to GitHub do not count!) or **upload**.