Unit: Critical 1: Mathematics and Statistics for Data Science
Instructor: Jasper Shuoyang Zheng

Generative AI Checklist

Have you read UAL Student Guide to Generative AI? If not, READ IT NOW! It provides details on how you may use Generative AI such as ChatGPT, Copilot, or any large language model in your assessments. Most importantly, it tells you what will constitute Academic Misconduct.

For the Math&Stats unit, we welcome the use of Copilot for learning and coding support, and we ask you to **think responsibly** of how you use it, and how it may be helpful for your learning.

- 🚫 **What you should never do:**
  - AI as shortcuts (e.g., copy and paste AI generated content without reading it)
  - Skipping your own judgements (e.g., accepting everything generated by AI without considering whether they are correct or not)
  - AI misuse (e.g., submit AI generated content and claim that it is your own work)

In practice, the four main principles are:

- ✅ **Alway Rubber Duck**
  `rubber-duck` is a chat mode for Copilot, tailored for the Math&Stats unit. It gives you full access to Copilot's capability, and it is adapted for learning support, adapted for content in this unit, and help you avoiding unintended misconduct. You should always use the `rubber-duck` chat mode for this unit.
  - Please refer to the Setup Copilot guide for how to set up Rubber Duck in VS Code.
- ✅ **Read, Review, and Adapt**
  You should always read your Copilot's full message, review what it has done to your workspace/code/notebook, and adapt its work if needed.
- ✅ **Keep Logs and Export Chat**
  You should keep logs of how you have used AI to reach your final outcome:

    - Logs can include answers to the following questions (you can find a template in the Example section below):
        - What did you ask the AI to do?
        - What did the AI attempt? What was added to the code/notebook by the AI?
        - What did you decide to keep, change, discard? What did you learn from this?
    - In the Time Constrained Assessment, you should also export a transcript of your chat history with Copilot. This can be done by `command + shift + p` (or `ctrl + shift + p` on Windows), type `Chat: Export Chat`.

- ✅ **Add a Generative AI Disclosure**
  You should always add a **Generative AI Disclosure** section at the end of your work to disclose the use of AI in your notebooks, codes, written texts. Adding comments by `#` to disclose AI generated content is also a common practice used by programmer.

    - In academic writting, the Cite Them Right website provides guidance on how to reference generative AI in work. Read: Generative AI (Harvard) Referencing Guide.

Examples

Example of a written **chat log** from a conversation:

> **What did you ask Copilot to do**:
>
> I asked it to calculate the correlation of temperature and latitude in the climate dataset.
>
> **What did Copilot do**:
>
> - It first broke down the task to "loading dataset", "select dataframe column", "calculate Pearson correlation coefficient", and "visualise relationship with a scatter plot".
> - It added a section in my notebook called "Correlation Analysis", the code first load the csv file, then selected the "temperature" and "latitude" column.
> - It attempted to run the code block, but there was an error.
> - It noticed that the error was due to a missing value in the csv file. Therefore it added another code block in my notebook to check and remove `nan` values.
> - Then it added the correlation calculation.
>
> **What did you decide to keep, change, discard?**
>
> - I deleted most of the redundant `print` functions.
> - I kept the part where `pd.read_csv()` was used to loading csv file, the `.isnull().sum()` and `dropna()` part where it removes empty values, and the `stats.pearsonr()` part to calculate correlation.
> - I also deleted the visualisation part as I don't need it at this point.

Example of a written **Generative AI Disclosure**:

> AI Usage Disclosure: The code in Section 3 of this notebook was created with assistance from Copilot (GPT5 mini). The content has been reviewed and edited by Jasper Zheng. For more information on the extent and nature of AI usage, please refer to the "GenAI-log.pdf" submitted.

Example of an **in-code AI disclosure**:

```
# Added by Copilot
# Prompt: Create a function that multiply two numbers.
def multiply(a, b):
    y = a * b
    return y
```

Checklist

For each of your sessions with Copilot, have you:

- [ ] Read through all messages from Copilot?
- [ ] Reviewed whether Copilot has added any new files to your workspace?
- [ ] Noticed key functions/operations it added to your code?
- [ ] Thought about whether these functions/operations are appropriate?
- [ ] Tested whether these functions/operations actually work, and do what you want them to do?
- [ ] Provided the **written chat log** describing how you used AI to reach your outcome?
- [ ] Answered the three questions in the template chat log.

For each notebook that you're going to submit, have you:

- [ ] Reviewed what Copilot did to your code/notebook?
- [ ] Provided in-code comments to disclose these contents created by AI?
- [ ] Provided a Generative AI Disclosure?

For the Time Constrained Assessment, have you:

- [ ] Included the exported chat transcript? This can be done by `command + shift + p` (macOS) or `ctrl + shift + p` (Windows), then type `Chat: Export Chat`.
- [ ] If you have multiple sessions with Copilot, make sure to export all of them.