SM-2302 Software for Mathematicians

Introduction & Getting Started

Drs. Haziq Jamil & Huda Ramli
Mathematical Sciences, Faculty of Science, UBD
https://github.com/sm2302-aug23

Semester I 2023/24

Admin

- Lecturer information


```
Dr. Haziq Jamil
Assistant Professor in Statistics
Room M1.09
haziq.jamil@ubd.edu.bn
```

```
Dr. Huda Ramli
Assistant Professor in Applied Mathematics
Room M1.04
huda.ramli@ubd.edu.bn
```


- **IMPORTANT: Read the syllabus.**
- Weekly contact hours
  - Lectures: Tuesdays 2:10–4.00 pm @ ICTC Lab 7
  - Labs: Fridays 2:10–4.00 pm @ ICTC Lab 7
- Be aware of schedule and important deadlines.
- Check Canvas regularly for announcements and course materials.

Module description

Mathematical software is what bridges higher mathematics to real world applications. On completing this module, students should be able to use `MATLAB` and `R` to effectively implement mathematical solutions to real world problems. They should also be able to produce publication-quality mathematical documents using LaTeX. This module provides the computing skills required for an applied mathematics final year project.

Contents

1. Learning `MATLAB` and `R` languages for mathematical applications.
2. `MATLAB` specific outomes:
   - Basic operations, programming, numerical techniques and root finding.
3. `R` specific outcomes:
   - Logic and types, data frames and matrices, data wrangling, and visualisations.
4. Preparation of report-style documents using LaTeX.
5. Version control and social coding using Git and GitHub.

Assessment

Take note that this module is assessed wholly (100%) by coursework.

Formative assessment

- Lab-based tutorials

Summative assessment

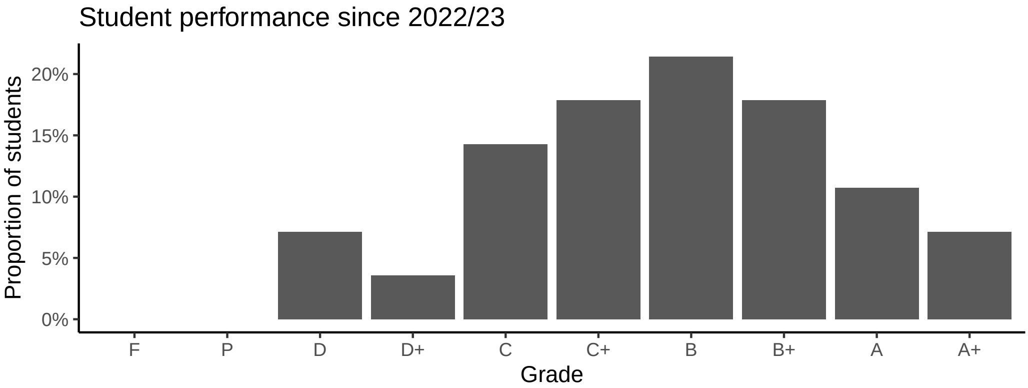
- **[10%]** 2 × online quizzes (R and MATLAB)
- **[5%]** 1 × Canvas discussion (Git & GitHub)
- **[20%]** 2 × mini individual assignments
- **[30%]** 2 × mini group assignments
- **[5%]** 2 × group peer review
- **[30%]** 1 × project assignment with written report

Schedule

Week	Topic	Instructor	Assessment
W01: 31/07 – 06/08	Introduction & Getting Started	HJ	
W02: 07/08 – 13/08	[R] Logic and types	HJ	
W03: 14/08 – 20/08	[R] Matrices and data frames	HJ	Quiz 1
W04: 21/08 – 27/08	[Git] Git and GitHub	HJ	Discussion 1
W05: 28/08 – 03/09	[R] The tidyverse	HJ	Individual 1
W06: 04/09 – 10/09	[R] Visualisations using ggplot	HJ	
W07: 11/09 – 17/09	[MATLAB] Basic operations	NHR	
18/09 – 24/09	Mid-semester Break		
W08: 25/09 – 01/10	[R] Peer review	HJ	Group 1
W09: 02/10 – 08/10	[MATLAB] Programming	NHR	
W10: 09/10 – 15/10	[MATLAB] Numerical techniques	NHR	Quiz 2
W11: 16/10 – 22/10	[MATLAB] Root-finding	NHR	Individual 2
W12: 23/10 – 29/10	[LaTeX] Typesetting reports	NHR	
W13: 30/10 – 05/11	[LaTeX] Beyond reports	NHR	
W14: 06/11 – 12/11	[MATLAB] Peer review / presentations	NHR	Group 2

Past student feedback

- *Fun and applied module*
- *Challenging if not have coding experience*
- *Useful for FYP and other modules*
- *Intensive and lots of work involved*



Use of AI, collaboration, sharing and code reuse

- All graded assignments must be your own individual work, *except* for the group assignments (these are expected to be collaborative in nature). **Otherwise, NO CODE SHARING among peers**.
- We are aware that the internet is a great resource. You may make use of any of these resources, but **you must explicitly <u>cite</u> where you obtained any code you have used directly, or gained inspiration from**.
- Any recycled code that is discovered and is not explicitly cited will be treated as plagiarism, regardless of source.
- The use of AI tools (including but not limited to ChatGPT) to produce solutions to any assignments is **<u>prohibited</u>**, and if caught using will be treated as plagiarism.

Plagiarism

Penalties for plagiarism

- A written, formal reprimand kept in Faculty records; and/or
- Resubmission of assignment; and/or
- Reduced assignment marks; and/or
- Fail grade for assignment.

Admin

Purpose of mathematical software

Getting started

Purpose of mathematical software

> *Software is essential for modelling, analysing and calculating numeric, symbolic, or geometric data.*

Generally speaking, mathematical software is very focused:

1. **Software calculator**: Performs simple mathematical operations.
2. **Computer algebra systems**: Designed to solve classical algebra equations and problems in human readable notation.
3. **Statistics**: Statistical analysis of data.
4. **Optimisation**: Selecting a best solution from a set of alternatives.
5. **Numerical analysis**: Numerical approximations for the problems of mathematical analysis.
6. etc.

Remark

While mathematical software produces useful solutions, they very often do not explain why the solutions are what they are.

How many primes are there?

How many primes are there?

Theorem 1 (Euclid's Theorem)

There are infinitely many primes.

How many primes are there?

Theorem 1 (Euclid's Theorem)

There are infinitely many primes.

- A prime number $p \in \{n \in \mathbb{N} \mid n > 1\}$ is divisible only by itself and 1.
- We might attempt to brute force the answer by writing a software loop.
- Can we prove this theorem by software?

How many primes are there?

Theorem 1 (Euclid's Theorem)

There are infinitely many primes.

- A prime number $p \in \{n \in \mathbb{N} \mid n > 1\}$ is divisible only by itself and 1.
- We might attempt to brute force the answer by writing a software loop.
- Can we prove this theorem by software?

```
INPUT n
  i := 2
  count := 0
  WHILE i <= n
    rem := n % i
    IF rem not equal to 0
      i := i + 1
      count := count + 1
    END IF
  END WHILE
OUTPUT count
```

Software affords us insight

Let $\pi(x)$ be the prime counting function defined to be the number of primes less than or equal to x, for any $x \in \mathbb{R}$. Can we intuit a good approximation of $\pi(x)$?

Software affords us insight

Let $\pi(x)$ be the prime counting function defined to be the number of primes less than or equal to x, for any $x \in \mathbb{R}$. Can we intuit a good approximation of $\pi(x)$?

A different (but related) question: How far apart are the prime numbers?

Software affords us insight

Let $\pi(x)$ be the prime counting function defined to be the number of primes less than or equal to x, for any $x \in \mathbb{R}$. Can we intuit a good approximation of $\pi(x)$?

A different (but related) question: How far apart are the prime numbers?

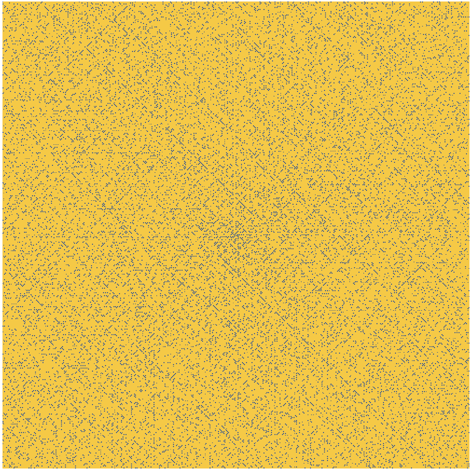
Define the *density* of primes as $\pi(x)/x$. This gives an idea of the distribution of primes up to x. It would be interesting to map this out.

Source code from
`https://github.com/johnistan/ulam-spirals-R`

21	22	23	24	25
20	7	8	9	10
19	6	1	2	11
18	5	4	3	12
17	16	15	14	13

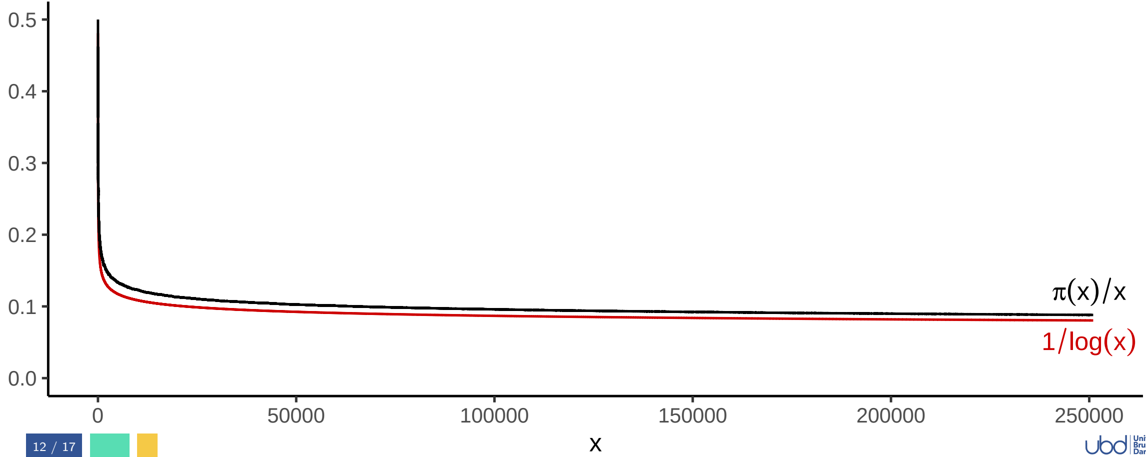
Ulam's spiral

- Prominent diagonal, horizontal and vertical lines containing large number of primes.
- Not unsurprising, as these correspond to certain prime-generating polynomials such as $x^2 - x + 41$ (Euler's).
- Nonetheless, connected to many unsolved areas of mathematics!
  - Riemann Hypothesis
  - Goldbach's conjecture
  - Twin prime conjecture
  - Legendre's conjecture



Does the density converge?

As $x \to \infty$, the prime density $\pi(x)/x$ diminishes at a slow rate. Reminiscent of an inverse logarithmic decrease!



The prime number theorem

- The *asymptotic* law of distribution of prime numbers states that

$$\lim_{x \to \infty} \frac{\pi(x)/x}{1/\log(x)} = \frac{\pi(x)}{x/\log(x)} = 1$$

The prime number theorem

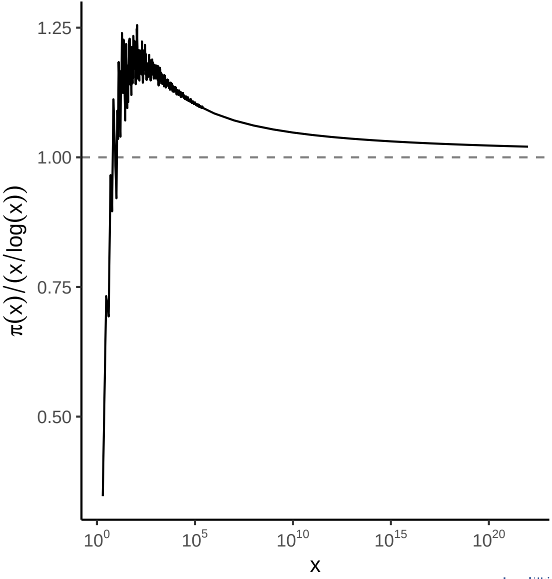
- The *asymptotic* law of distribution of prime numbers states that

$$\lim_{x\to\infty} \frac{\pi(x)/x}{1/\log(x)} = \frac{\pi(x)}{x/\log(x)} = 1$$

- From this, we have

$$\pi(x) \sim \frac{x}{\log(x)}$$

- We now have an approximation for the prime counting function, which improves as x increases. In particular, $\lim_{x\to\infty} x/\log(x) = \infty$.



Using software

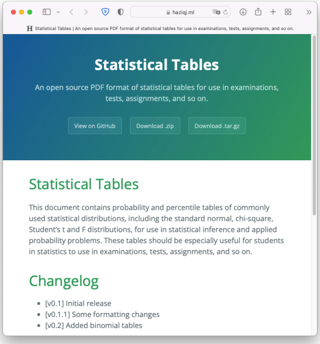
$$\begin{tikzcd}[column sep=huge] \text{Mathematics} \arrow[r, bend left, "\textit{ideation}"] & \texttt{Software} \arrow[l, bend left, "\textit{results}"] \end{tikzcd}$$

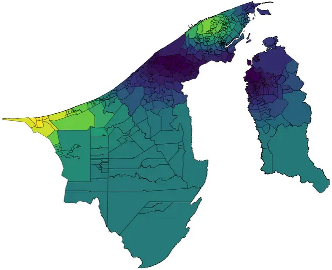
Use software as a tool to...

- Explore and visualise ideas
- Confirm ideas numerically
- Communicate results

Beyond this course

Static websites served on GitHub; 3-D plots and animation; Plotting GIS shape files and maps; Reproducible research (`knitr`); Text processing and analysis; Web and social media scraping; Creating R packages; Web APIs; Parallel computing; Optimisation; Mathematical and statistical modelling





Admin

Purpose of mathematical software

Getting started

- Instructions
- Software overview

Instructions

IMPORTANT

Check Canvas for detailed instructions regarding software installation and sign up procedures.

Important points:

- Use UBD e-mail in most cases to obtain Education Benefits
- Pick a suitable username (one that you won't be embarassed to use in a few years time!)
- Practice safe and secure passwords
- When using Lab PCs, best to create a personal folder and keep all your work files in there.
- Using your own laptops is fine. Mind your cables! Avoid tripping hazards.
- Recommended to use USB drives (make sure they're clean!) or some cloud service (Dropbox, Sharepoint, Google Drive, etc.)

Software overview

1. MATLAB–more details in NHR's sessions.
2. RStudio Desktop
    - RStudio is installed on campus computers.
    - It is free to install on your personal computers–`https://www.rstudio.com/products/rstudio/download/`
    - You may also need to install the `R` language too, depending on your system. Do a Google search for 'R Windows download' or similar.
3. Git, github.com and GitHub Desktop
    - Please sign up for an account at github.com/signup using your UBD e-mail.
    - You will be invited to join the course organization (`sm2302`) in due course.
    - Assignments will be distributed and collected via GitHub.
4. Overleaf.com
    - Please sign up for an account at `https://www.overleaf.com/register`