يونيبرسيتي بروني دارالسلام
UNIVERSITI BRUNEI DARUSSALAM

SM-2302 Software for Mathematicians

Introduction & Getting Started

Drs. Haziq Jamil & Huda Ramli

Mathematical Sciences, Faculty of Science, UBD

`https://sm2302.github.io`

Semester I 2022/23

Overview

Admin

- Lecturer information

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

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

- **IMPORTANT: Read the syllabus.**
- Weekly contact hours
    - Lectures: 2 hours in ICTC Lab 7 on Tuesdays 2:10 PM
    - Tutorials: 2 hours in ICTC Lab 7 on Saturdays 2:10 PM
- 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

- **[20%]** 4 × online quizzes
- **[20%]** 2 × mini individual assignments
- **[30%]** 2 × mini group assignments
- **[30%]** 1 × project assignment with written report

Assessment

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

Formative assessment

- Lab-based tutorials

Summative assessment

- **[20%]** 4 × online quizzes
- **[20%]** 2 × mini individual assignments
- **[30%]** 2 × mini group assignments
- **[30%]** 1 × project assignment with written report

A note on teams

- You may self-sign up to form teams (max in a group is 4).
- Peer evaluation after completion.
- Everyone is expected to contribute **equal effort**.

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). **Do not share your code, otherwise**.
- We are aware that the internet is a great resource. You may make use of any of these resources, but **you must explicitly cite where you obtained any code your directly use or use as inspiration in your solutions**.
- Any recycled code that is discovered and is not explicitly cited will be treated as plagiarism, regardless of source.

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). **Do not share your code, otherwise**.
- We are aware that the internet is a great resource. You may make use of any of these resources, but **you must explicitly cite where you obtained any code your directly use or use as inspiration in your solutions**.
- Any recycled code that is discovered and is not explicitly cited will be treated as plagiarism, regardless of source.

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.

Schedule

Week	Topic	Instructor	Assessment
W01: 01/08 – 07/08	Introduction & Getting Started	NHR & HJ	
W02: 08/08 – 14/08	[MATLAB] Basic operations	NHR	
W03: 15/08 – 21/08	[MATLAB] Programming	NHR	
W04: 22/08 – 28/08	[R] Introduction to R and Rstudio	HJ	Quiz 1
	[Git] Git and GitHub	HJ	
W05: 29/08 – 04/09	[MATLAB] Numerical techniques	NHR	Quiz 2
W06: 05/09 – 11/09	[MATLAB] Root-finding	NHR	Individual 1
W07: 12/09 – 18/09	[R] Logic and types	HJ	
19/09 – 25/09	Mid-semester Break		
W08: 26/09 – 02/10	[MATLAB] Peer review / presentations	NHR	Group 1
W09: 03/10 – 09/10	[R] Matrices and data frames	HJ	Quiz 3
W10: 10/10 – 16/10	[R] The tidyverse	HJ	
W11: 17/10 – 23/10	[R] Visualisations using ggplot	HJ	Individual 2
W12: 24/10 – 30/10	[LaTeX] Typesetting reports	HJ	
W13: 31/10 – 06/11	[LaTeX] Beyond reports	NHR	Quiz 4
W14: 07/11 – 13/11	[R] Peer review	HJ	Group 2

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 <u>why</u> 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 $p \in \mathbb{N}$ 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 $p \in \mathbb{N}$ 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?

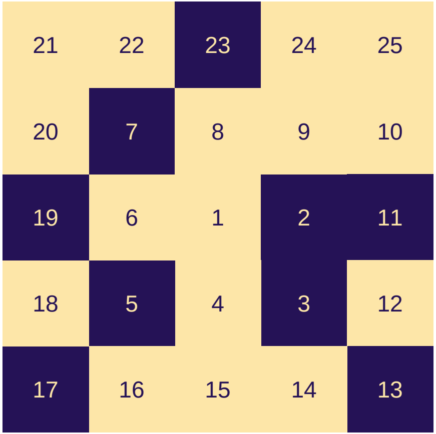
Software affords us insight

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?

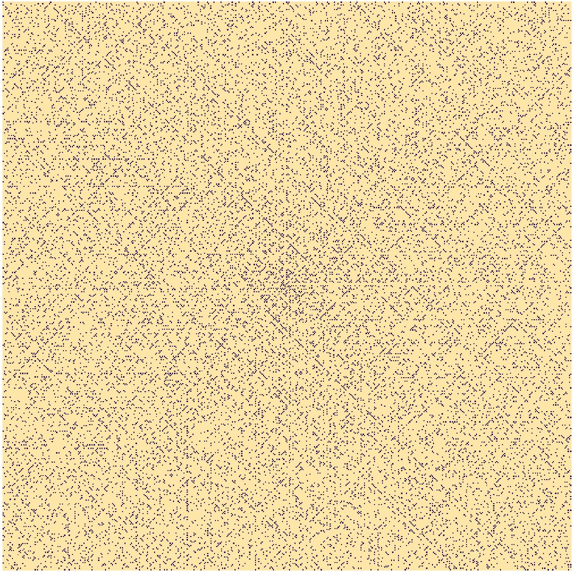
Define the *density* of primes as $\pi(x)/x$. This gives an idea of the distribution of primes up to x. It would be interesting to map this out.

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



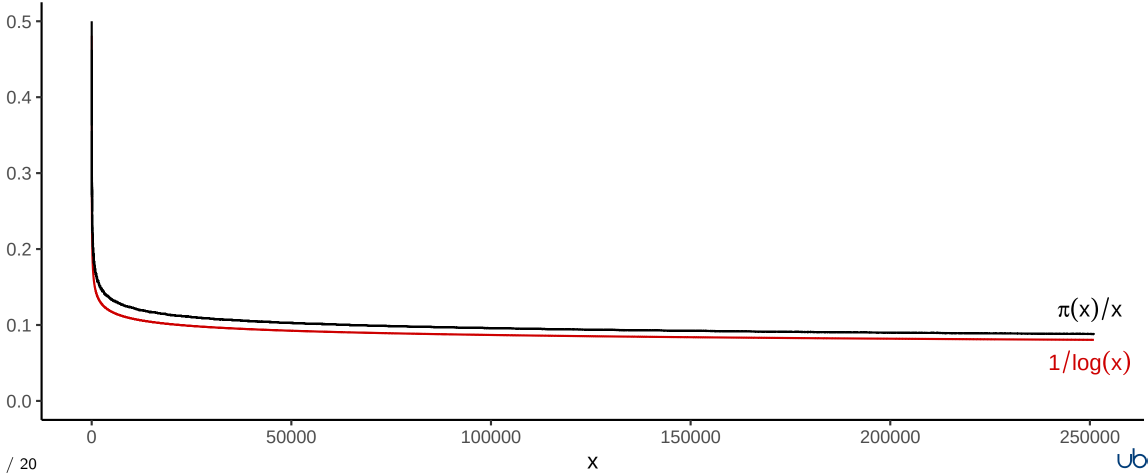
Ulam's spiral

- 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

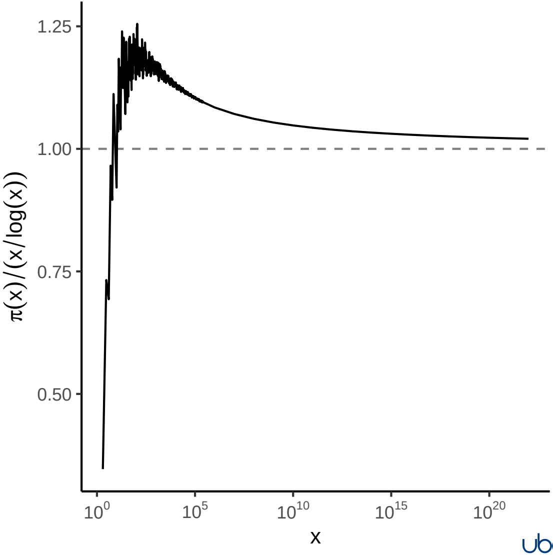
- The *asymptotic* law of distribution of prime numbers states that

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

- 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

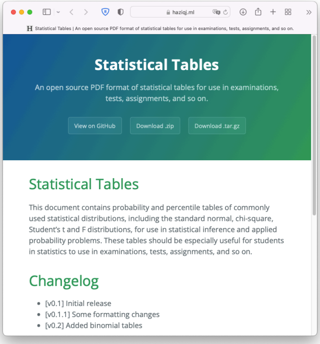
$$\begin{tikzcd}
\text{Mathematics} \arrow[r, bend left, "\textit{ideation}"] & \texttt{Software} \arrow[l, bend left, "\textit{results}"]
\end{tikzcd}$$

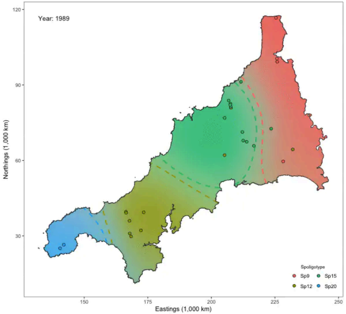
Use software as a tool to...

- 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





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.
- 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 the upcoming slides.
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`

MATLAB

- MATLAB is a high-level language and interactive environment for numerical computation, visualization and programming.
  - Analyze data
  - Develop algorithms
  - Create models and applications
- We will reinforce some calculus concepts and its applications using MATLAB, such as
  - Numerical differentiation
  - Numerical Integration
  - Root-finding methods

Software

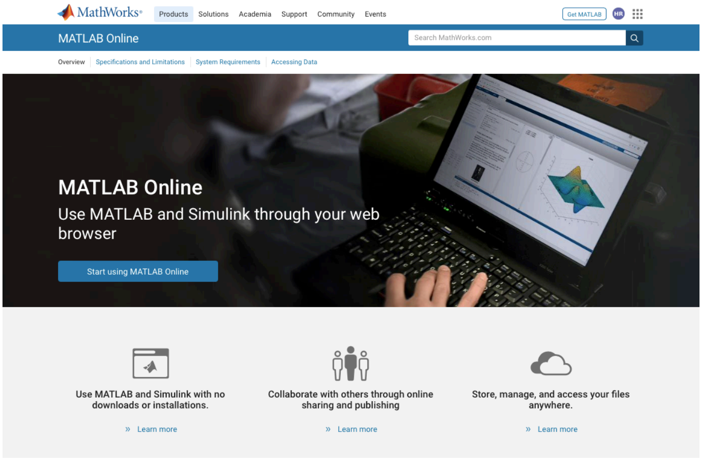
MATLAB is installed in the campus computer labs. However, if you wish to work from home or on your laptop, you can either

1. use MATLAB Online on your web browser; or
2. install MATLAB on your personal computer.

MATLAB Campus-wide suite

To install or use MATLAB on your web browser, you need to create a Mathworks account using your UBD e-mail. You can access the UBD campus-wide suite using your mathworks account. Please refer to the MATLAB individual CWL installation guide document.

MATLAB Online



https://matlab.mathworks.com

MATLAB Graphical Interface

- Command window
- Workspace
- Current folder
- Command history
- Editor