# HIGH PERFORMANCE RESEARCH COMPUTING

## Micro-credentials in Research Computing

09/30/2022

High Performance
Research Computing
DIVISION OF RESEARCH

# Outline

- Overview of Microcredentials
- Details about Specific Courses
- Demonstrations of Online Course Elements

# Overview of Microcredentials

# Background

- The High Performance Research Computing Group has many years of experience teaching computer skills.
  - Short courses
  - Workshops
  - Summer Camps
  - 1-on-1 User Support
- We are excited to announce that we are ready to take our expertise to the next level.

# Short Courses

- Short courses provide small-scale learning opportunities targeting specific skills.
- Short courses fill gaps in coursework for traditionally non-computing degree programs.
- Online, asynchronous courses developed with support from Texas A&M Engineering Studio for Advanced Instruction and Learning (eSAIL).

TEXAS A&M UNIVERSITY
Engineering Studio for
Advanced Instruction & Learning

# Research Skills

- Exercises and topics designed for researchers
  - E.g., graduate students
- Coding skills for specific disciplines and purposes
- Training in standard workflows and common practices
  - Especially useful for High Performance Computing

*Not just another generic tutorial!*

# Microcredentials

- Microcredentials are rewards for completion of small-scale learning goals, recognized by academia and industry.
- Supported by Texas A&M Engineering Experiment Station Education Generations (TEES EDGE)
- This program is part of the professional and continuing education credit system.
  - Outside the university course credit system

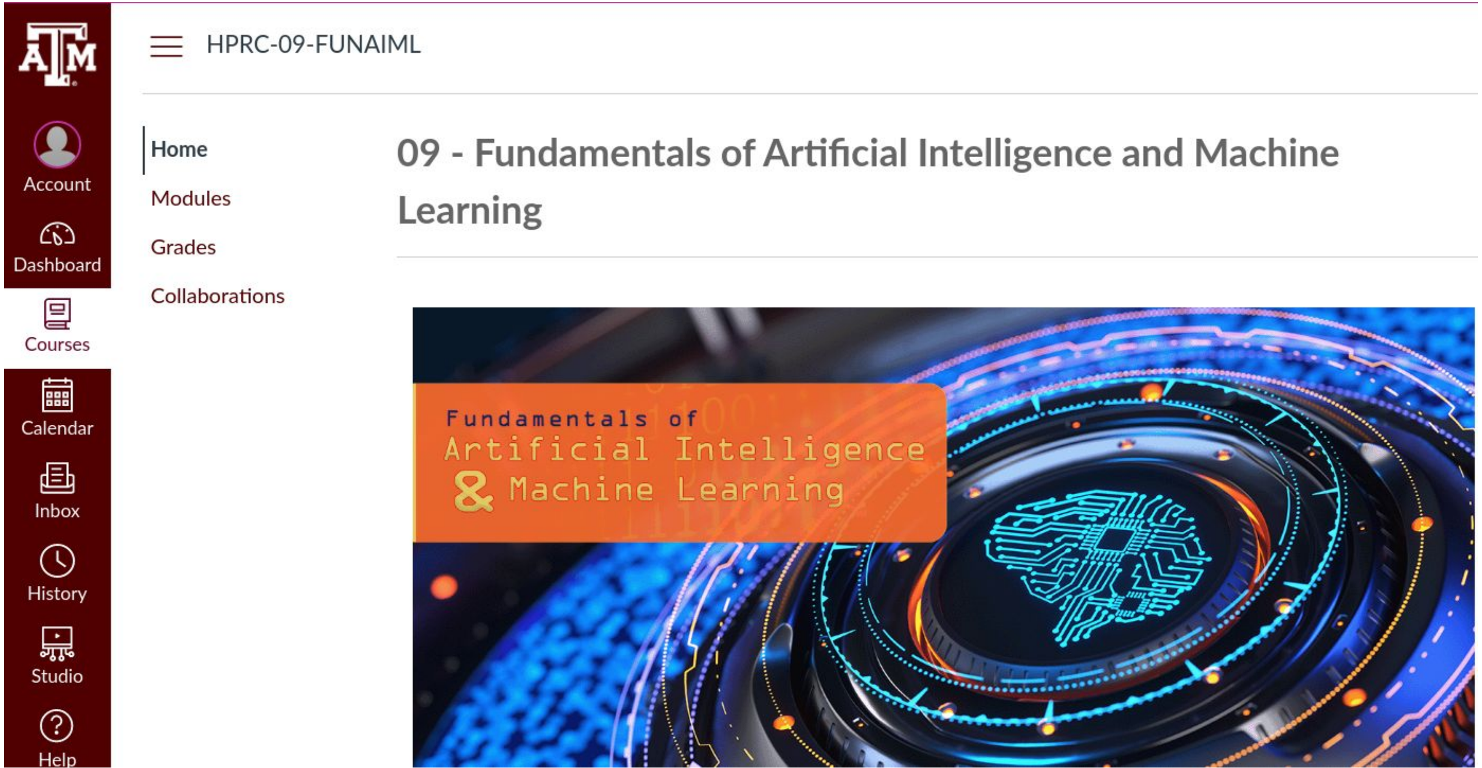
TEES EDGE
Professional Education

# Audience

- This program will eventually be publicly available
  - Currently: only TAMU members, by invitation

*We invite you to help us test our courses!*

# Canvas Learning Management System

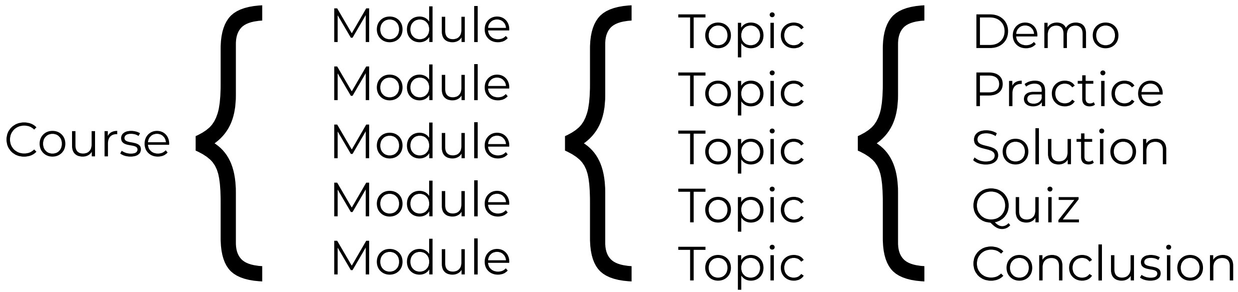


HPRC-09-FUNAIML

09 - Fundamentals of Artificial Intelligence and Machine Learning

# Online Course Structure

Online Courses are organized hierarchically.



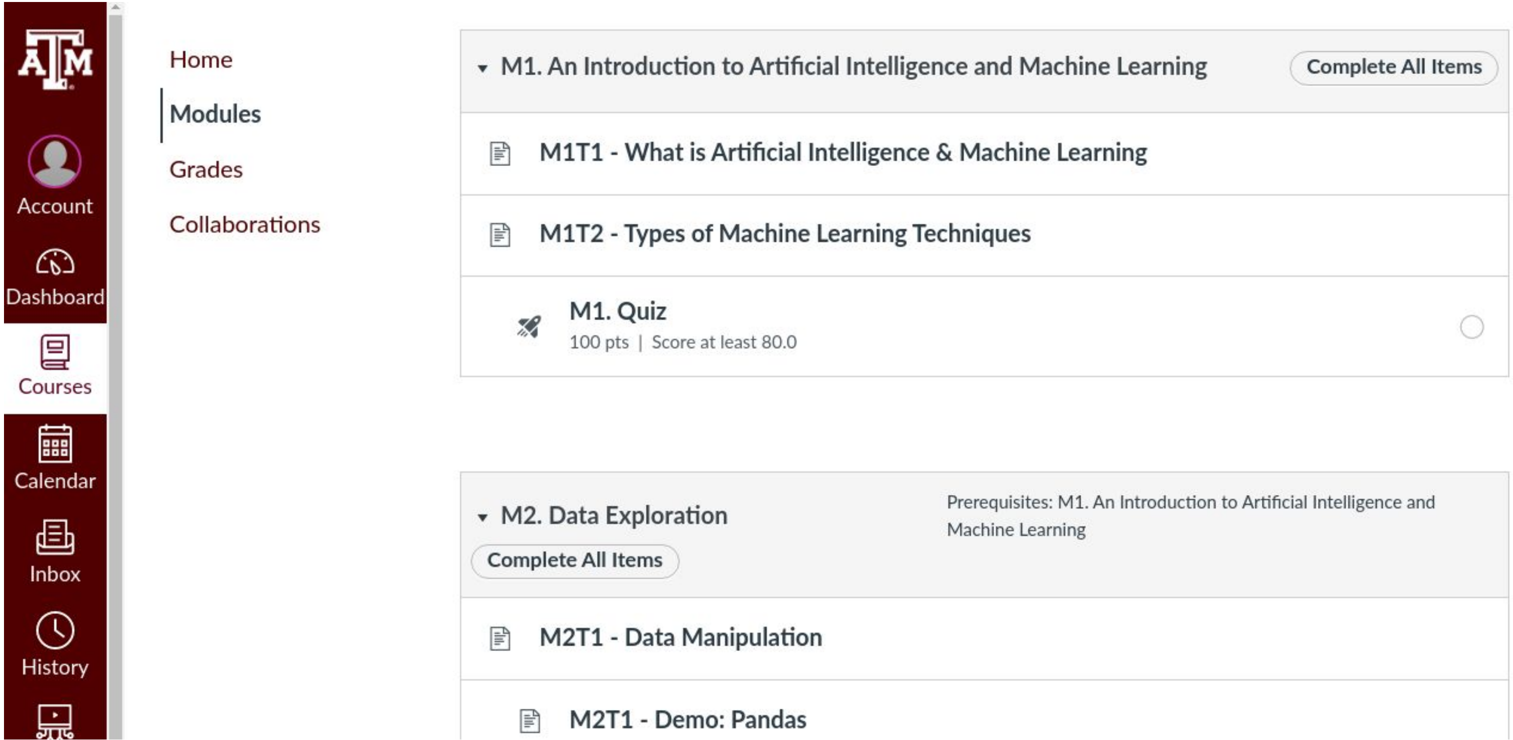
Course

Module
Module
Module
Module
Module

Topic
Topic
Topic
Topic
Topic

Demo
Practice
Solution
Quiz
Conclusion

# Example Course Hierarchy



▾ M1. An Introduction to Artificial Intelligence and Machine Learning — Complete All Items

- M1T1 - What is Artificial Intelligence & Machine Learning
- M1T2 - Types of Machine Learning Techniques
- M1. Quiz — 100 pts | Score at least 80.0 ○

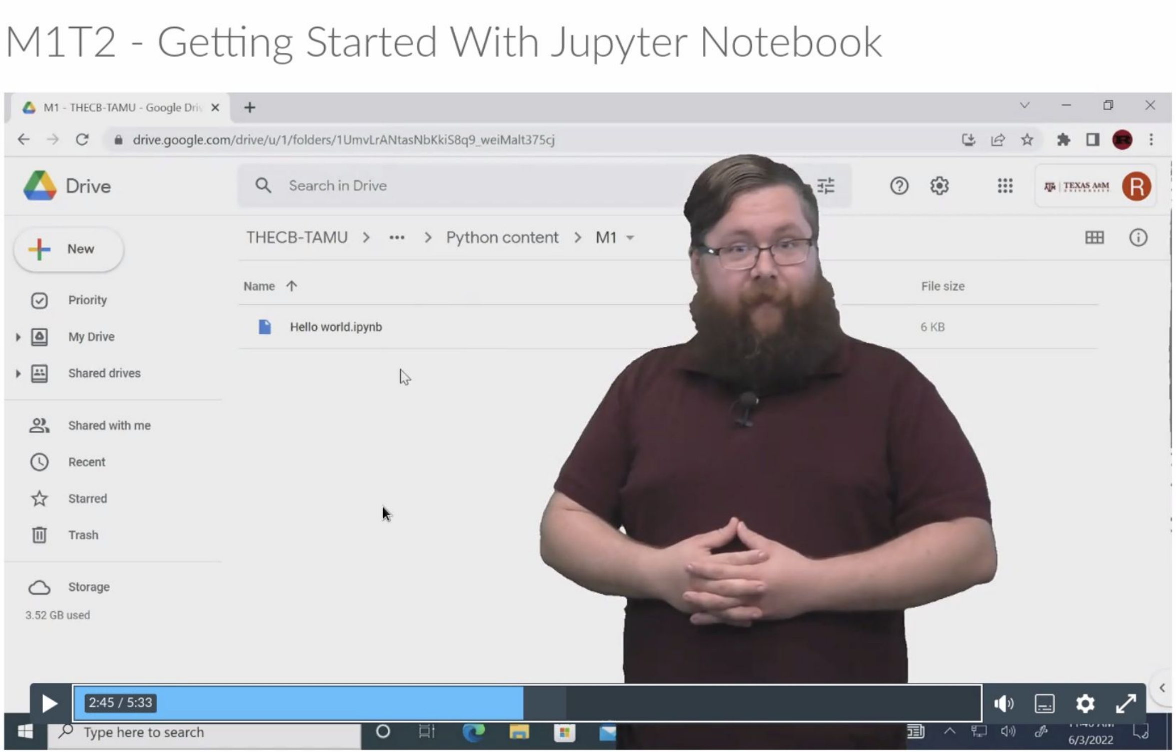
▾ M2. Data Exploration — Complete All Items — Prerequisites: M1. An Introduction to Artificial Intelligence and Machine Learning

- M2T1 - Data Manipulation
- M2T1 - Demo: Pandas

# Demo, Practice, Solution

- Each Topic has a cadence of course materials
  - Demo (a video lecture) 5-10 minutes
  - Practice 10-20 minutes
  - Solution (a video lecture) 5-10 minutes
- The student stays maximally engaged with minimal wait between activities.

# Example Demo Video



M1T2 - Getting Started With Jupyter Notebook

# Quizzes for Completion

- Each module has at least one quiz.
  - “about 5” questions selected at random from a pool of questions.
  - Retries are allowed, but the questions will vary.
- Passing the module quiz is required to proceed to the next module.
- Some courses also have a final quiz.
- Passing all quizzes (≥ 80%) is required to earn your micro-certificate.

# Table of Courses (1/2)

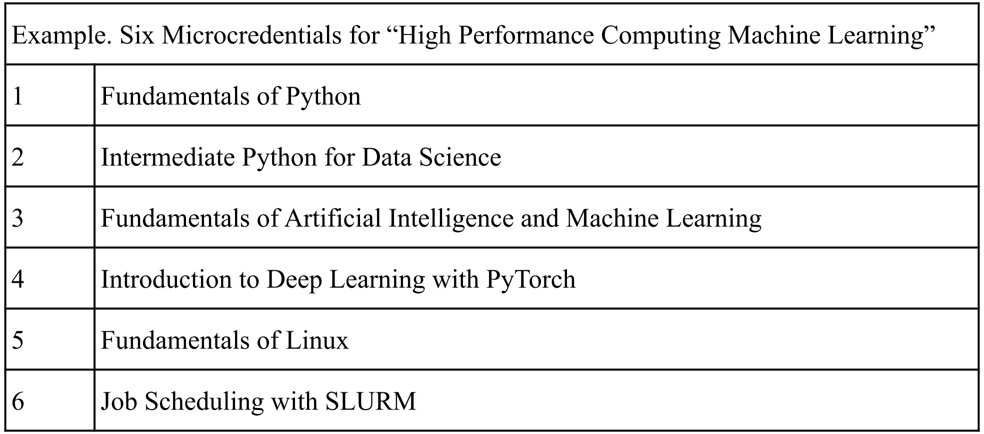
| Theme | Course Title | CIP CODE(s) | Duration | Delivery Mode |
|---|---|---|---|---|
| Cybersecurity | Fundamentals of Cybersecurity | 11.1003 | 0.3 PDU | Asynchronous |
| Coding | Fundamentals of Python | 11.0201 | 1 PDU | Both |
| | Intermediate Python for Data Science | 11.0202, 30.71 | 1 PDU | Both |
| | Advanced Python for Data Science | 11.0202, 30.71 | 0.5 PDU | Both |
| | Fundamentals R Programming | 11.0201 | 1 PDU | Both |
| | Intermediate R Programming | 11.0202 | 1 PDU | Both |
| | GPU Programming with CUDA | 30.3001 | 0.3 PDU | Live |
| AI/ML | Fundamentals of Artificial Intelligence and Machine Learning | 11.0102 | 0.3 PDU | Both |
| | Introduction to Deep Learning with TensorFlow | 11.0804, 11.0202 | 0.3 PDU | Both |
| | Introduction to Deep Learning with PyTorch | 11.0804, 11.0202 | 0.3 PDU | Both |
| | Using Scikit-Learn for Artificial Intelligence and Machine Learning | 11.0804, 11.0104, 11.0202 | 0.3 PDU | Both |

# Table of Courses (2/2)

| Theme | Course Title | CIP CODE(s) | Duration | Delivery Mode |
|---|---|---|---|---|
| Bioinformatics | RNA-seq and Differential Expression | 11.0104, 11.0401, 26.1103 | 0.3 PDU | Both |
| | Short Variant Discovery | 11.0104, 11.0401, 26.1103 | 0.3 PDU | Both |
| | Metagenomics | 11.0104, 11.0401, 26.1103 | 0.3 PDU | Both |
| | ChIP-seq | 11.0104, 11.0401, 26.1103 | 0.3 PDU | Both |
| Linux | Fundamentals of Linux | 11.0201 | 0.3 PDU | Both |
| | Linux for Administrators | 11.1006, 11.1001 | 0.3 PDU | Live |
| Cloud Computing | Job Scheduling with SLURM | 11.0103 | 0.3 PDU | Live |
| | Containers and Orchestration | 11.0103 | 0.3 PDU | Live |
| | Introduction to Cloud and Cluster Computing | 11.0103 | 0.3 PDU | Live |
| | Parallel Computing Using OpenMP | 11.0201 | 0.3 PDU | Asynchronous |
| | Parallel Computing Using MPI | 11.0201 | 0.3 PDU | Asynchronous |

# Stackable Courses

- Some short courses have prerequisites
- Stack microcredentials to build your own program



| Example. Six Microcredentials for "High Performance Computing Machine Learning" | |
|---|---|
| 1 | Fundamentals of Python |
| 2 | Intermediate Python for Data Science |
| 3 | Fundamentals of Artificial Intelligence and Machine Learning |
| 4 | Introduction to Deep Learning with PyTorch |
| 5 | Fundamentals of Linux |
| 6 | Job Scheduling with SLURM |

# Details about Specific Courses

# Cybersecurity

One course:

- Fundamentals of Cybersecurity

# Fundamentals of Cybersecurity

- This course covers fundamental concepts in cybersecurity

- M1 - Introduction and Security Trends
    - The security problem
    - Threats to security
    - Attributes of actors
    - Security trends
    - Targets and attacks
    - Approaches to cybersecurity

# Fundamentals of Cybersecurity

- M2 - General Security Concepts
  - Security basics
  - Security tenets
  - Security approaches
  - Security principes
  - Security models

- M3 - The Role of People in Security
  - Social engineering
  - Poor security practices
  - People as a security tool

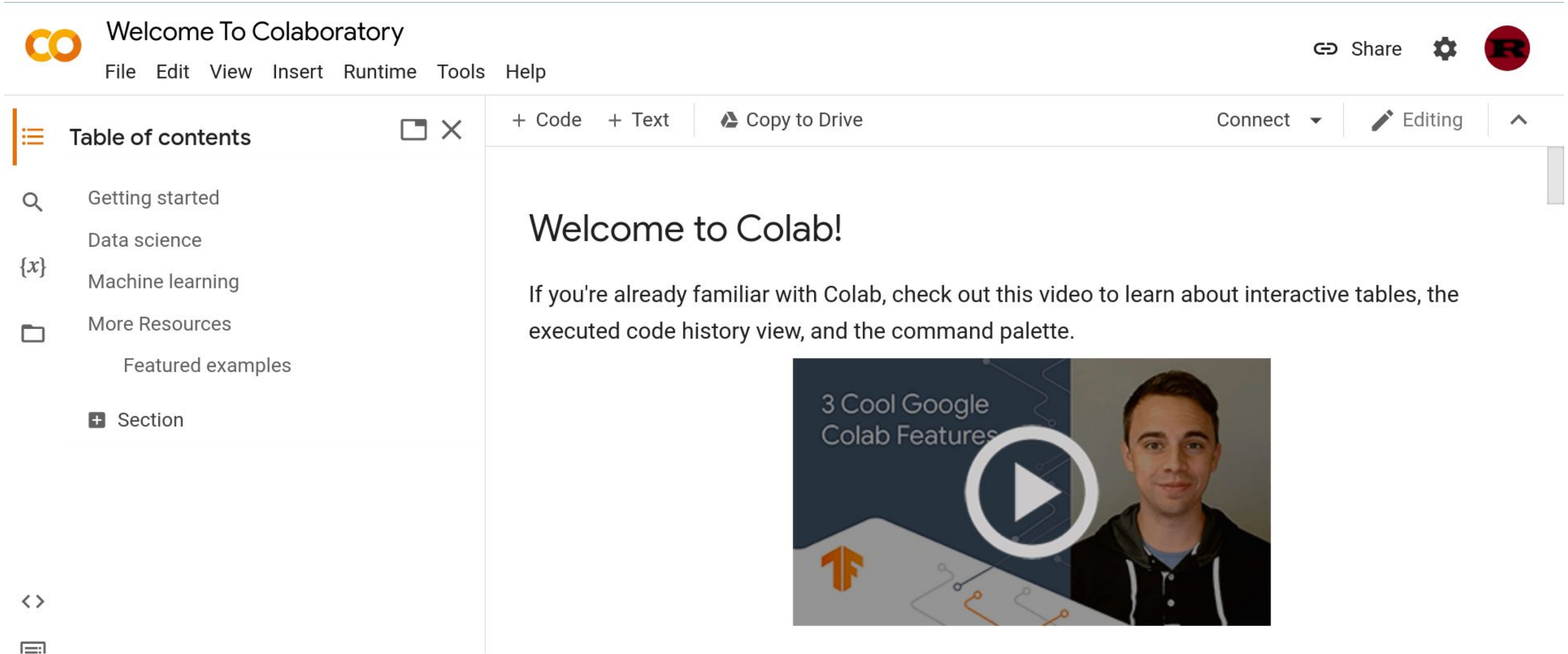
# Fundamentals of Cybersecurity

- M4 - Types of Attacks and Malicious Software
    - Avenues of Attack
    - Malware
    - Attacking Computer Systems
    - Advanced Persistent Threats
    - Tools
    - Auditing

# Python Programming

- Three courses:
  - Fundamentals of Python
  - Intermediate Python for Data Science
  - Advanced Python for Data Science
- Each course is a *prerequisite* for the next.
- Learning takes place using the Google Colab integrated development environment.
- Topics and exercises are selected to be relevant for scientific research applications.

# Google Colaboratory



Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

# Fundamentals of Python

- 10 hours
- This course covers the most important core components of Python programming at the introductory level.
- Students will learn fundamental programming concepts such as variables, data structures, flow control, and object-oriented programming.

# Fundamentals of Python

- Using Google Colaboratory
- Variables
- Data Types
- User Input
- Operations
- Control structures
- Errors and Files
- Lists
- Dictionaries
- Classes
- Modules

# Intermediate Python for Data Science

- 10 hours
- This course covers a selection of scientific programming tools commonly used in Python programming at the intermediate level.
- Students will learn research techniques such as manipulating and visualizing data, exploring functions, modeling, and retrieving data from the internet.
- *Prerequisite: Fundamentals of Python*

# Intermediate Python for Data Science

- Arrays with NumPy
- Plotting with Matplotlib
- Mathematics with SciPy
- Data Tools with Pandas
- Web Scraping

# Advanced Python for Data Science

- 3 hours
- This course covers a selection of scientific programming tools commonly used in Python programming at the advanced level.
- Students will learn research techniques such as handling multidimensional data, and parallelization.
- *Prerequisite: Intermediate Python for Data Science*

# Advanced Python for Data Science

- Arrays with Xarray
- Parallelization with Dask

# Artificial Intelligence and Machine Learning

- Four courses:
  - Fundamentals of Artificial Intelligence and Machine Learning
  - Using Scikit-Learn for Artificial Intelligence and Machine Learning
  - Introduction to Deep Learning with TensorFlow
  - Introduction to Deep Learning with PyTorch
- The four courses are *independent* of each other.
- *Recommended* prerequisite: Intermediate Python
- Learning takes place using the Google Colab integrated development environment.

# Fundamentals of Artificial Intelligence and Machine Learning

- Introduction to AI/ML
    - What is AI/ML and their Relationship
        - AI and ML Description
        - Relationships
    - Terminology
        - Data Formats
        - Types of Data
            - Labeled Data
            - Unlabeled Data
        - What is training?
        - What is testing?
        - Common types of Learning

# Fundamentals of Artificial Intelligence and Machine Learning

  - Types of ML techniques
    - Supervised learning
      - Regression
      - Classification
    - Unsupervised learning
      - Clustering
- Data Exploration
  - Data Manipulation with Pandas
    - Demo
    - Practice
  - Data Visualization with Matplotlib
    - Demo
    - Practice

# Fundamentals of Artificial Intelligence and Machine Learning

- Application of ML techniques
  - Demo 1: Predict House Prices
  - Practice 1: Linear Regression
  - Demo 2: Classify Handwritten Digits
  - Practice 2: Classification
  - Demo 3: Build a Clustering Model
  - Practice 3: Clustering
- Final Assessment

# Using SciKit-learn for Artificial Intelligence and Machine Learning

- Introduction to ML Algorithms
    - Linear Regression
        - Guided Practice
    - Logistic Regression
        - Guided Practice
    - Support Vector Machine
        - Guided Practice
    - K-Means Clustering
        - Guided Practice

# Introduction to Deep Learning with TensorFlow

- Overview of AI/ML/DL
  - What is Deep Learning and Its Advantages
  - Learning Principle of DL
  - Convolution
  - Pooling
  - Convolutional Neural Network (CNN)
- TensorFlow
  - TensorFlow Introduction
  - TensorFlow Keras API
  - Demo 1: Get Started with TensorFlow
  - Demo 2: Classify Handwritten Digits with TensorFlow

# Introduction to Deep Learning with PyTorch

- Overview of AI/ML/DL
  - What is Deep Learning and Its Advantages
  - Learning Principle of DL
  - Convolution
  - Pooling
  - Convolutional Neural Network (CNN)
- PyTorch
  - PyTorch Introduction
  - Major Components
  - Demo 1: Get Started with PyTorch
  - Demo 2: Classify Fashion MNIST with PyTorch

# R Programming

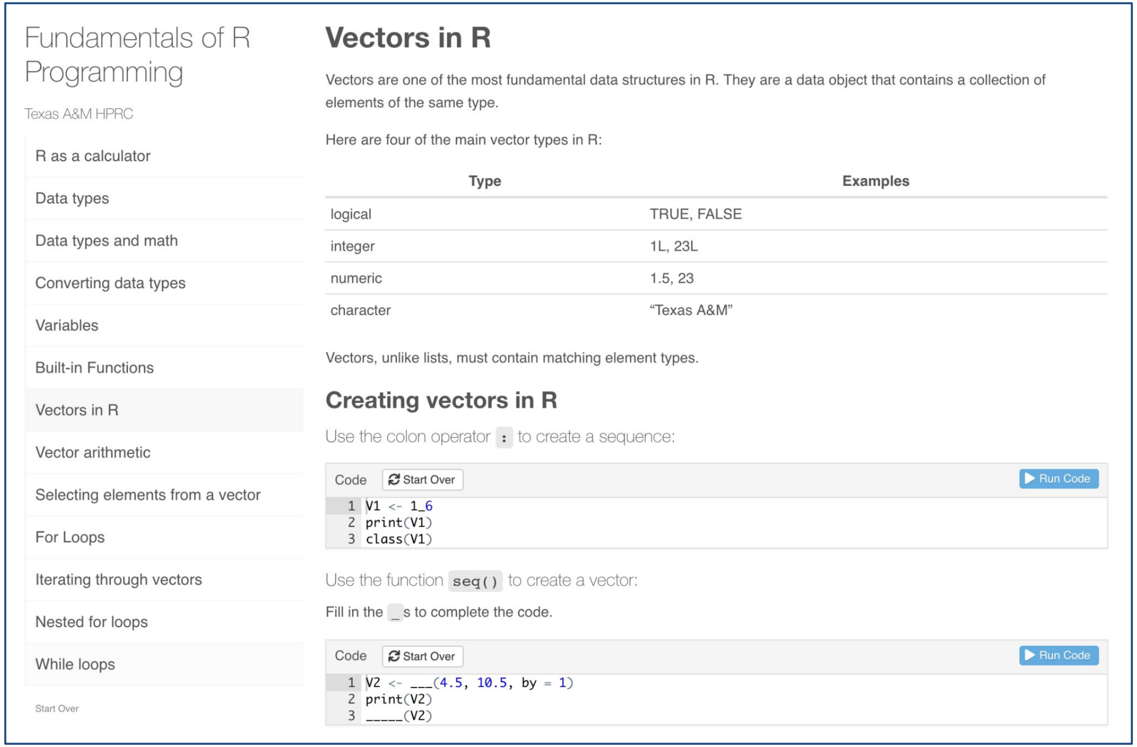
- Two courses:

1. Fundamentals of R Programming
2. Intermediate R Programming

- The first is a *prerequisite* for the second.

# Fundamentals of R Programming

- 6 hours
- RStudio
- Topics Covered:
  - Data types
  - Variables
  - Built-in Functions
  - Vectors
  - Loops



Fundamentals of R Programming

Texas A&M HPRC

- R as a calculator
- Data types
- Data types and math
- Converting data types
- Variables
- Built-in Functions
- Vectors in R
- Vector arithmetic
- Selecting elements from a vector
- For Loops
- Iterating through vectors
- Nested for loops
- While loops

Start Over

## Vectors in R

Vectors are one of the most fundamental data structures in R. They are a data object that contains a collection of elements of the same type.

Here are four of the main vector types in R:

| Type | Examples |
|---|---|
| logical | TRUE, FALSE |
| integer | 1L, 23L |
| numeric | 1.5, 23 |
| character | “Texas A&M” |

Vectors, unlike lists, must contain matching element types.

### Creating vectors in R

Use the colon operator `:` to create a sequence:

```
V1 <- 1_6
print(V1)
class(V1)
```

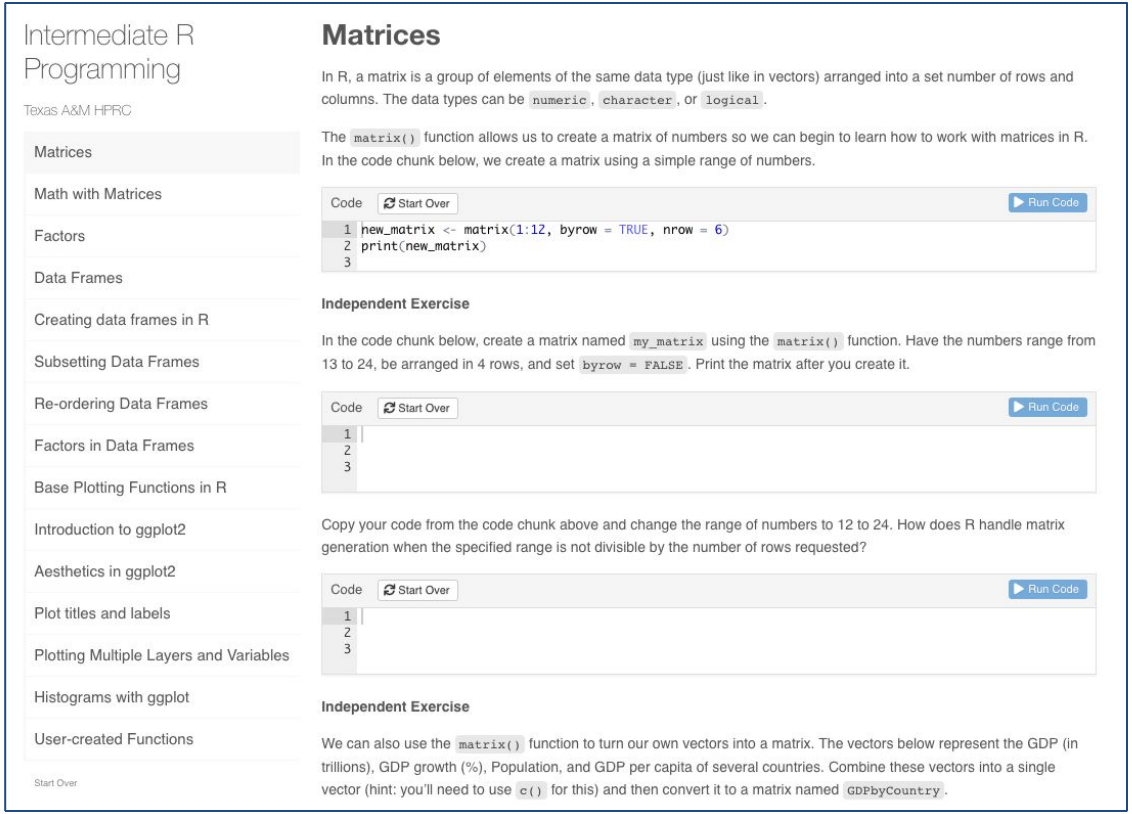
Use the function `seq()` to create a vector:

Fill in the `_`s to complete the code.

```
V2 <- ___(4.5, 10.5, by = 1)
print(V2)
_____(V2)
```

# Intermediate R Programming

- 6 hours
- RStudio
- Topics Covered:
    - Matrices
    - Factors
    - Data frames
    - Base plotting
    - ggplot2
    - User-created functions



## Matrices

In R, a matrix is a group of elements of the same data type (just like in vectors) arranged into a set number of rows and columns. The data types can be `numeric`, `character`, or `logical`.

The `matrix()` function allows us to create a matrix of numbers so we can begin to learn how to work with matrices in R. In the code chunk below, we create a matrix using a simple range of numbers.

```r
new_matrix <- matrix(1:12, byrow = TRUE, nrow = 6)
print(new_matrix)
```

**Independent Exercise**

In the code chunk below, create a matrix named `my_matrix` using the `matrix()` function. Have the numbers range from 13 to 24, be arranged in 4 rows, and set `byrow = FALSE`. Print the matrix after you create it.

Copy your code from the code chunk above and change the range of numbers to 12 to 24. How does R handle matrix generation when the specified range is not divisible by the number of rows requested?

**Independent Exercise**

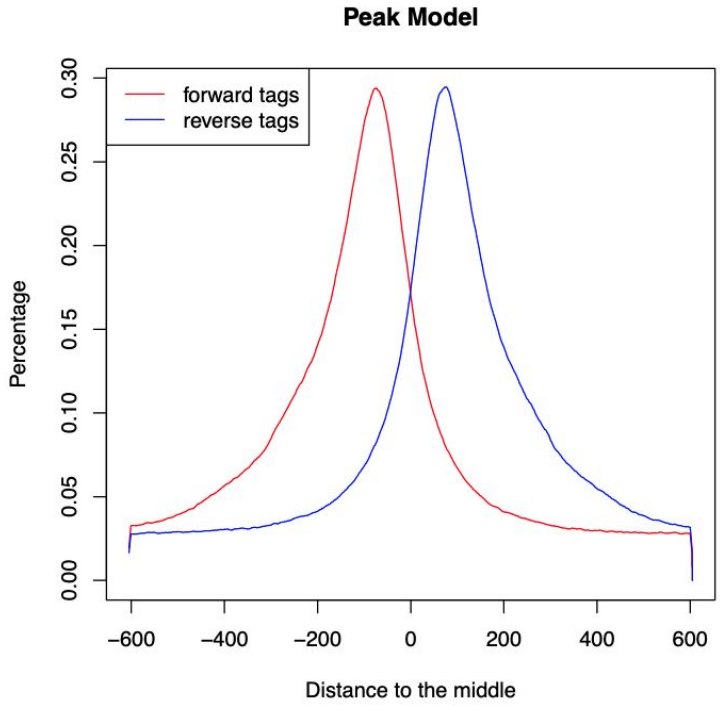
We can also use the `matrix()` function to turn our own vectors into a matrix. The vectors below represent the GDP (in trillions), GDP growth (%), Population, and GDP per capita of several countries. Combine these vectors into a single vector (hint: you'll need to use `c()` for this) and then convert it to a matrix named `GDPbyCountry`.

# Biology Applications

- Four courses:
  - Introduction to ChIP-seq
  - Short Variant Discovery
  - Introduction to Metagenomics
  - RNA-seq and Differential Expression
- These four courses are *independent* of each other

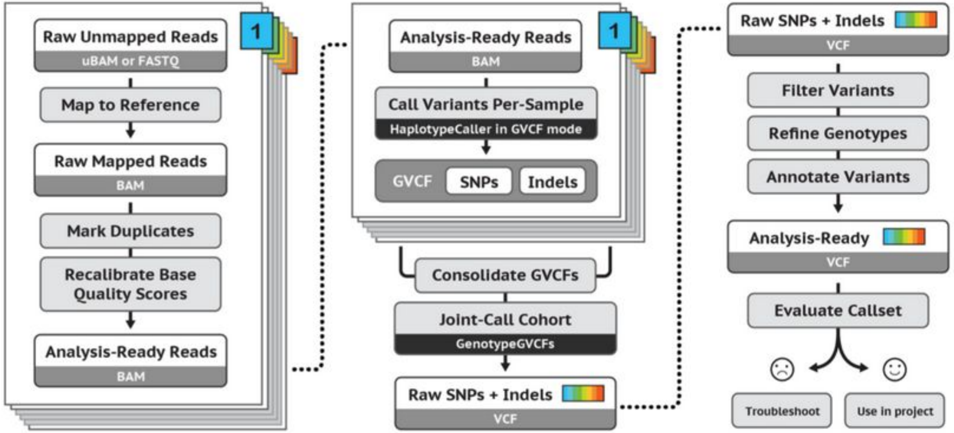
# Introduction to ChIP-seq

- 3 hours
- ChIP-seq experiment overview
- Commonly used software
- Bioinformatics on the Grace Cluster
- Full ChIP-seq pipeline
  - Library QC and Trimming
  - Read mapping
  - ChIP-QC
  - Peak calling



# Short Variant Discovery

- 5 hours
- Commonly used software
- Bioinformatics on the Grace cluster
- Step-by-step through GATK
  - Library QC and Trimming
  - Alignment and processing
  - Per-sample variant calling
  - Cohort sample calling
  - Variant filtering



https://gatk.broadinstitute.org/hc/en-us/articles/360035535932-Germline-short-variant-discovery-SNPs-Indels-

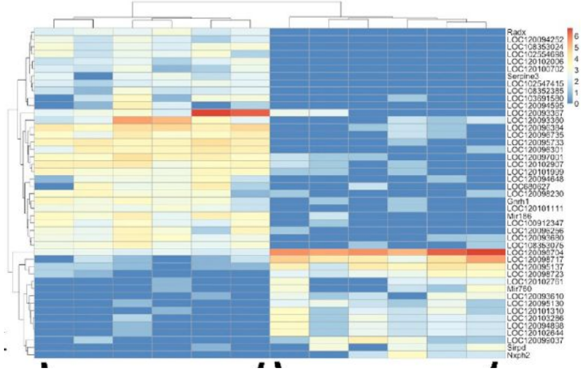
# Introduction to Metagenomics

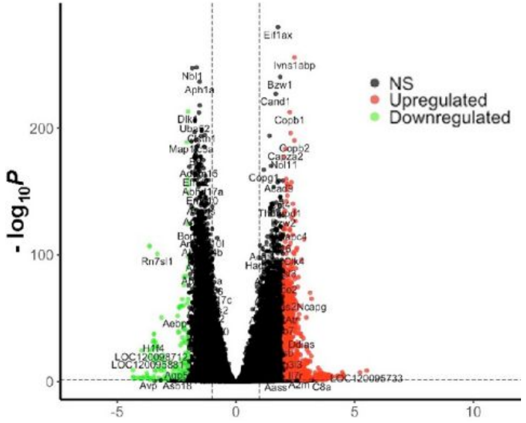
- 3 hours
- Metagenomics sequencing strategies
- Commonly used software
- Full pipeline for marker gene analysis (16s) with QIIME2
  - QIIME2 terminology/format
  - Importing data
  - Denoising and filtering
  - Diversity analyses
  - Taxonomic classification

https://qiime2.org/

# RNA-seq and Differential Expression

- 4 hours
- Discuss commonly used RNA-seq software
- Experimental Design
- Full Differential Expression Pipeline
  - Illumina library processing
  - Short-read mapping
  - Count generation
  - Differential expression analysis in R
  - Data visualization in R





Orzabol et al. 2022

# Demonstrations of Online Course Elements

*time permitting*

# Acknowledgements

This work was supported by the

- National Science Foundation (NSF) award number 1925764, "CC* Cyberteam: SWEETER"
- NSF award number 2019129, "MRI: FASTER"
- NSF award number 1730695, "CyberTraining: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students”
- NSF award number 2019136, "CC* BRICCs: Building Research Innovation at Community Colleges"
- NSF award number 1829799, "Cybertraining: CMS3"
- NSF award number 2112356, "ACES - Accelerating Computing for Emerging Sciences"

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

https://hprc.tamu.edu

HPRC Helpdesk:

help@hprc.tamu.edu
Phone: 979-845-0219

Help us help you. Please include details in your request for support, such as, Cluster (Faster, Grace, Terra, ViDaL), NetID (UserID), Job information (Job id(s), Location of your jobfile, input/output files, Application, Module(s) loaded, Error messages, etc), and Steps you have taken, so we can reproduce the problem.