QTM 531: COMPUTING II

Contact Hours: Two 1.25-hour or one 2.5-hour session weekly, plus 4.5 hours of regular, out-of-class work required as preparation for in-class work

Credit Hours: 3

Instructor:	Xxx
Semester	Spring 20xx
Meeting Time and Place:	xxx
Office:	xxx
Office Hours:	xxx
Email/Contact:	xxx
Course Website:	xxx
TA:	xxx

Course Objectives

This class is the second sequence of the two computing courses connecting to QTM 530 Computing I. Assuming that students know how to explore and manipulate data, and do the basic programming, this course will focus on gaining building blocks for programming related to data analysis and machine learning. In addition, the class will introduce practical concepts relevant for reproducible research with big data. By the end of the course, students are expected to (1) fluently reshape data into the most convenient form for analysis, (2) know how to implement methods related to data analysis, (3) know how to implement algorithms in machine learning, (4) know how to implement statistical methods and machine learning algorithms using cloud, (5) know how to make a research reproducible by understanding from the development and source control to the deployment. Students would primarily write code in Jupyter/IPython notebooks. Most of the computing exercises will be based on Python.

Prerequisites

- QTM 530 Computing I

Required Textbook

- [IML] Introduction to Machine Learning with Python, by Andreas C. Müller and Sarah Guido, O'Reilly
- [PD] Python for Data Analysis, by Wes McKinney, O'Reilly
- [IPE] Introduction to Python for Econometrics, Statistics and Data Analysis, by Kevin Sheppard, (https://www.kevinsheppard.com/files/teaching/python/notes/python_introduction_2019.pdf)
- [GLM] statsmodels.org – GLM notebook (https://www.statsmodels.org/stable/examples/notebooks/generated/glm.html)
- [FPP] Foundations of Python Programming (https://runestone.academy/runestone/books/published/fopp/index.html)
- [PE] Python for Everybody, by Charles R. Severance, (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)
- [HDS] How to think like a Data Scientist, (https://runestone.academy/runestone/books/published/httlads/index.html)
- [BB] Bash for Beginners [BB], by Machtelt Garrels, (https://www.tldp.org/LDP/Bash-Beginners-Guide/Bash-Beginners-Guide.pdf)
- [IL] Introduction to Linux [IL], by Machtelt Garrels, (https://www.tldp.org/LDP/intro-linux/html/)
- [ACP] AWS Cloud Practitioner Essentials [ACP], 2nd Ed, (https://www.aws.training/Details/Curriculum?id=27076&scr=path-cp)

Class Requirements

Grades will be based on

- homework assignments (45%)
- a midterm exam (20%)
- a final exam (30%)
- class participation (5%)

Homework

The homework assignment consists of 7 computer-based problem sets. Usually each homework would due before the class. Any assignment submitted after the due date/time will be considered 0 points. To accommodate unexpected circumstances, your lowest homework grade will be automatically dropped at the end of the semester. Working together on the homework assignments is encouraged, but you must write your own answers. It is highly recommended that you make your solo effort on all the problems before consulting others.

Exams

There are is a midterm exam and a final exam. **No collaboration** is allowed on the exams. There would be **no make-up exam**. However, with Emory approved excuses, the missed exam would be weighted toward the final exam.

Honor Code

All students enrolled at Emory are expected to abide by the Emory College Honor Code. Any type of academic misconduct is not allowed which includes 1) receiving or giving information about the content or conduct of an examination knowing that the release of such information is not allowed and 2) plagiarizing, whether intentionally or unintentionally, in any assignment. For the activities that are considered to be academically dishonest, refer to the Honor Code:
http://catalog.college.emory.edu/academic/policies-regulations/honor-code.html.

Disability Accommodations

If you are seeking classroom accommodations or academic adjustments under the Americans with Disabilities Act, you are required to register with Office of Accessibility Services (OAS), http://accessibility.emory.edu/. Once registration is finalized, students must request accommodation needs to be communicated or facilitated. Students are expected to give two weeks' notice of the need for accommodations for any class activities including the exams. For more information, please see http://accessibility.emory.edu/students/new-to-oas/registering.html. Please make sure to contact me with the relevant letter at the beginning of the semester.

Tentative Course Schedule

Part I: Advanced Computing Techniques

Week 1: Essential computer literacy and Linux Foundations with Shell script

Essential computer literacy *Reading [BB] Chapter 1 & 3*

- Binary
- Characters and unicode
- High vs low level programming languages
- Compiled vs interpreted languages

Linux Foundations *Reading [IL]*

- Navigating file system (cd, ls, and related commands)
- Manipulating files (cp, rm, and related commands)
- Redirection operators (| and >)

- Installing software
- Archiving with tar
- Using ssh

Introduction to scripting *Reading [BB] Chapter 4*
- Executing shell scripts
- Writing shell scripts
- Shell variables (export and PATH)

Homework 1

Part II: Introduction to Python

Week 2: Python Basics

Basic Python Syntax **(Review)** *Reading [PD] Chapter 3.1 & [FPP] 2 & 7*
- The Python interpreter (interactive shell, python command, notebooks)
- Python's data model (objects, values, and types)
- Numbers (int, bool, float)
- Iterables (str, list, tuple, dict)

Data manipulation and Visualization **(Review)** *Reading [PD] Chapter 4 & 5*
- Requests
- Pandas
- Numpy
- Bokeh

Week 3-4: Programming in Python

Introduction to Programming *Reading [PD] Chapter 3.2 & [FPP] 4 & 12*
- Programming in a Jupyter notebook
- Programming in an IDE
- Regular expressions
- List comprehensions
- Functions (built-in, user-defined)
- Modules
- PEP 8

Homework 2

Programming Principles *Reading [FPP] 5.4*
- Classes in Python
- Attributes and methods
- OOP summary (encapsulation, inheritance, and polymorphism)

Part III: Data Analysis with *scikits.statsmodels*

Week 5: Analysis Based on Linear Models

Reading [IPE] Chapter 20
Data analysis with OLS estimator
Data analysis with randomized experiment

Homework 3

Week 6: Analysis Based on Non-Linear Models

Data analysis with limited dependent variable *Reading [GLM]*

- Logit model
- Poisson model

Homework 4

Week 7

Review & Midterm Exam

Part IV: Machine learning algorithm with *scikit-learn* and *TensorFlow*

Week 8-9 Supervised Learning

Example 1: Decision Tree (*scikit-learn*) *Reading [IML] Chapter 1.4 & 2.1-2.2 & 2.3.5*

- Training and Test Sets
- Train Decision Tree
    - Regression
    - Classification *Reading [IML] Chapter 2.4*
- Model Evaluation *Reading [IML] Chapter 5.1-5.2 & 5.3.2*

Example 2: Neural Networks (*TensorFlow*) *Reading [IML] Chapter 2.3.8*

Homework 5

Week 10: Unsupervised Learning

Example: Understanding the data by K-means clustering (*scikit-learn) Reading [IML] Chapter 3.1-3.3 & 3.5*

Homework 6

Part V: Using Databases with Python

Week 11 Database System Foundations

Reading [PE] Chapter 15

Basic Query Language

- Create single table with Create, Read, Update, and Delete (CRUD)

Relational SQL

- Representing a Data Model in Tables
- Join

Many-to-many relationships

Part VI: Development and Production

Week 12-13:

Development *Reading [HDS] 4.1*

- Documentation and design
- Managing package version conflicts
- Virtual environments

- Conda
- Docker and containers

Source code control using Github

- Git
- Repositories
- Pull requests
- Collaborators

Deployment *Reading [HDS] 6.1*

- RESTful APIs
- Public cloud (Amazon Web Services, Google Cloud Platform)

Various online resources including:
https://guides.github.com/introduction/git-handbook/
https://aws.amazon.com/getting-started/
https://cloud.google.com/docs/overview

Homework 7

Part VII: Big data and Performance Optimization

Week 14: Big data and performance optimization

Reading [ACP]

Cloud computing

- Virtual machines and public cloud services for distributed computing
- Installing R and RStudio on cloud virtual machines
- Managed Jupyter Notebook services (AWS Sagemaker, GCP AI Platform)

Parallelism, multithreaded applications and concurrency

- Dask (parallelism for analytics in Python)
- Datashader (Visualization packages for large data)

Graphics processing units (GPU)

- Options for GPU computing on public cloud
- RAPIDS (GPU Accelerated libraries for data science)