ISOM 3390 Business Programming in R – Summer 2020

Instructor: QIAN, Bingjie (PhD candidate)

Contact: bqianaa@connect.ust.hk

Office: LSK 4063

Office hours: By appointment

TA (for labs): MAN, Ka Kuen (Josh)

Contact: imjosh@ust.hk

Office: LSK 4049C

Office hours: By appointment

Class Schedule (See Sections 9 &10 for more information)

Lecture: Mon & Wed & Fri (July 3, 6, 8, 10, 13, 15, 17, 20, 22, 24, 27, 29, 31)

Time: 1:30-4:20pm

Lab: Mon & Wed & Fri (July 3, 6, 8, 10, 13, 15, 17, 20, 22, 24, 27, 29, 31)

Time: 4:30pm-5:20pm

Course Website

Updates of the course materials (including lecture notes, videos, exercises, and assignments) and other information will be posted on Canvas (https://canvas.ust.hk/). Please check the course website regularly for updates.

---

1. Course Overview

In the era of big data, in order to fuel their decision making, firms need to analyze massive amounts of data in more efficient ways. Business applications and analytics relying only on basic spreadsheets and prepacked software tools are no longer adequate, and implementing sophisticated algorithms with custom programs starts to prevail. In addition, businesses are spending more time capturing data from various sources and curating the data before applying advanced analytic techniques. As such, mastering a programming language that helps in accomplishing all these tasks is crucial for students aiming for data applications or business analytics jobs.

With its extensive data visualization capabilities and continuously growing libraries, R is widely considered *the broadest analytical platform in the field of business analytics*. This course can prepare you with R programming skills for putting analytics and modeling techniques into use by working with cases of emerging business applications, including data visualization, Web scraping, text analytics, social network analysis, etc.

2. Course Goals and Objectives

At the end of this course, students will be able to:

- Understand generic programming language concepts in R
- Know how to obtain data from a variety of sources and tidy data for downstream analysis tasks with R
- Understand the basic principles of constructing data graphics and be familiarized with the plotting systems and visualization features in R
- Understand how to write R scripts and use various R packages for business applications
- Use R Markdown to write reports that includes R code and the code's automatically-generated output

3. Prerequisites

This course is *not* an introductory programming course. An understanding of essential programming concepts (data types and structures, control flow, functions, etc.) is a necessity to this course.

4. Lecture Notes, Textbooks, and Supplemental Readings

Before each class, the instructor will post the lecture notes (including in-class exercises) on Canvas, which outline the primary material for the class.

There are no specific textbooks required for this course, but you can easily find many useful resources online, for example, online books *Hands-on Programming with R* and *R for Data Science*.

Other reference books on R programming include:

- *The R Cookbook*, by Paul Teetor
- *The R Graphics Cookbook*, by Winston Chang
- *The Art of R Programming: A Tour of Statistical Software Design*, by Norman Matloff
- *Data Manipulation with R*, by Phil Spector
- *Software for Data Analysis: Programming with R*, by John Chambers (advanced book)

5. Grading

The grade breakdown for this course is as follows.

Components	Percentage of the Grade
A. In-class Exercises (13)	20%
B. Lab Exercises (13)	20%
C. Assignments (2)	30%
D. Final Project (group)	30%

6. Lectures and In-Class Exercises

During lecture sessions, students will learn the programming language concepts and skills covered in the lecture notes. In-class exercises will be given during lecture sessions to help student understand the concepts better and get hands-on practice.

Lecture sessions will be delivered via *live* Zoom meetings at https://hkust.zoom.us/j/99650431367 at scheduled times, unless an alternative mode is announced. Participating in live meetings is highly recommended. However, students can also watch the recorded videos to learn if they cannot participate in live meetings. The recorded videos are available upon request.

Each in-class exercise is to be submitted by *11:59am* on the next day of the corresponding lecture.

7. Labs

During the lab sessions, students will get hands-on practice with lecture concepts by working on assigned lab activities. Therefore, lab participation is an essential part of the learning process.

Lab sessions will be delivered via *live* Zoom meetings at https://hkust.zoom.us/j/94956022516 at scheduled times, unless an alternative mode is announced. The TA will be present to introduce the activities and to answer any questions you may have. Tasks may include but are not limited to: running or modifying code from the lecture, pair coding, or completing short coding exercises. Similar with lecture sessions, recorded videos are available upon request.

Each lab exercise is to be submitted by *11:59am* on the next day of the corresponding lab.

8. Assignments and Final Projects

There are two *individual* assignments, each possibly comprising conceptual questions to be answered and hands-on tasks. The due date of each assignment will be announced upon its release on Canvas.

No final exam will be administered. Instead, you will be assigned to small groups (3 at maximum) to work on a final project. You will select project topics provided by the instructor. Each group will cooperate on writing code, documenting it, and writing a report. One component (10% out of 30%) of your final project grade will be based on your teammates' assessment of your contribution to the project.

Rubrics

- Correctness: Deductions resulting from mistakes will be made at the discretion of the grader.
- Knitting: -5% deduction if the Rmd file you submit does not knit correctly (i.e., if there are errors and no HTML file is produced when the grader attempts to knit your Rmd file). If your Rmd file fails to knit, you will be contacted by the grader and will be given 24 hours to resubmit your work. You will need to trace the source of the error(s) and correct it.
- Style: Coding style is important. You will receive a deduction of up to 10% if you do not adhere to good coding style. Your code is considered to have a good coding style if:
    - good, consistent coding style
    - appropriate use of variables
    - appropriate use of functions
    - good commenting
    - good choice of variable names
    - appropriate use of inline code chunks

Late policy

Turn in your work early if there is any uncertainty about your ability to turn it in at the due time. Submissions up to 24 hours late will have their grade reduced by 25%; those up to 48 hours late will have their grade reduced by 50%. They will not be accepted for credit after two days.

Collaboration

You are encouraged to discuss in-class exercise and assignment problems with your fellow students. However, the work you submit must be your own. You must acknowledge in your submission any help received on your assignments. That is, you must include a comment in your homework submission that clearly states the name of the student, book, or online reference from which you received assistance.

Submissions that fail to properly acknowledge help from other students or non-class sources will receive no credit. Copied work will receive no credit. Any and all violations will be reported to Heinz College administration.

All students are expected to comply with the HKUST policy on academic integrity. This policy can be found online at http://ugadmin.ust.hk/integrity/student-1.html.

What constitutes plagiarism in a coding class?

The course collaboration policy allows you to discuss the problems with other students, but requires that you complete the work on your own. Every line of text and line of code that you submit must be written by you personally. You may not refer to another student's code, or a "common set of code" while writing your own code. You may, of course, copy/modify lines of code that you saw in lecture or lab.

You may find a discussion from the Computer Science and Engineering Department at the University of Washington helpful in understanding the bounds of the collaboration policy.

9. Tentative Lecture Schedule

Date	**Lecture Topic**	**Remark**
July 3	Topic 1: Course Introduction • Introduction to R and data science • Overview of planned topics • Course mechanics • RStudio • RMarkdown Topic 2: Data Structures • Basic data types • Vectors and factors	
July 6	Topic 2: Data Structures • Matrices and arrays • Lists • Data frames	
July 8	Topic 3: Control Structures • Conditional statements (conditionals) • Iterative statements (loops)	
July 10	Topic 4: Functions • Creating functions • Calling functions • Environments and scoping rules	
July 13	Topic 4: Functions • Environments and scoping rules Topic 5: Loop Functions • The apply family: lapply, sapply	Asg. 1 Release
July 15	Topic 5: Loop Functions • The apply family: mapply, apply, tapply • The plyr package: ddply	

July 17	Topic 6: Data Wrangling: A Tidy Approach • Tidy data, tidyverse • tibbles • tidyr for data tidying	
July 20	Topic 6: Data Wrangling: A Tidy Approach • The pipe operator • dplyr for data manipulation	
July 22	Topic 7: Base Plotting • The generic plotting function • The painter model • Other high-level plotting functions	
July 24	Topic 8: ggplot2 Plotting • Grammar of graphics • Data, geoms, and aesthetic mapping • Other grammatical elements	Asg. 1 Due Asg. 2 Release
July 27	Topic 9: String Operations • stringr for string operations • Regular expressions • Working with tidyr and dplyr	
July 29	Topic 10: Web Scraping • HTML and CSS basics • rvest for extracting tagged data • Dynamic web scraping	
July 31	Topic 11: Text Analytics • Text analytics with tidy text format • Sentiment analysis • Tokenizing and n-grams Course conclusion Final project requirements	Asg. 2 Due

10. Tentative Lab Schedule

Lab	**Date**	**Lab Topic**
1	July 3	R markdown, data type, operators, vectors, and factors
2	July 6	Exploring data frames
3	July 8	Using control structures
4	July 10	Writing and calling functions
5	July 13	Practicing advanced looping and basic loop functions

6	July 15	Practicing advanced loop functions
7	July 17	Data Wrangling with dplyr and tidyr I
8	July 20	Data Wrangling with dplyr and tidyr II
9	July 22	Plotting with R base graphics system
10	July 24	Plotting with ggplot2
11	July 27	Splitting and querying with regexes
12	July 29	Web scraping
13	July 31	Text analytics and sentimental analysis