ISOM3390 Business Programming in R – Winter 2020

Instructor: Dr. JIA, Jia (Justin)

Contact: justinjia@ust.hk

Office Hours: By appointment

TA: Mr. WONG, Ka Chau (Charles)

Contact: imcharles@ust.hk, 2358-7638

Office: LSK 4065

Office Hours: By appointment

Class Schedule (refer to the detail in section 12 and 13)

Lecture:

Mon. & Wed. & Fri (Jan. 3, 6, 8, 10, 13, 15, 17, 20, 22, 24)

Morning session: 10:00 – 11:50

Afternoon session: 13:00 – 14:50 (13:00 – 13:50 on Jan. 24)

Lab:

Mon. & Wed. & Fri (Jan. 3, 6, 8, 10, 13, 15, 17, 20, 22, 24)

15:00 – 15:50 (14:00 – 14:50 on Jan. 24)

Course Website

Updates of the course materials (including videos, notes, and exercises) and other information will be posted on the course website. Please check the course website regularly for updates.

---

1. Course Overview

In the era of Internet of Things (IoT) and big data, in order to fuel their decision making, firms need to analyze massive amounts of data for idiosyncratic problems 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 and has formal prerequisites: ISOM3230 and ISOM3360. An understanding of essential programming concepts (data types and structures, control flow, functions, etc.) is a necessity to this course.

4. Teaching Approach

In this course, we will adopt the flipped-classroom strategy that engages students in an active learning experience. It proves to be a better approach to move students from novice to skilled programmers than the traditional instructional one. Besides, it also provides students with opportunities to develop essential workplace skills such as critical thinking, written and oral communications, teamwork, and lifelong learning.

In this course, students are required to watch online lectures on designated topics in the morning session of every class day. Lecture videos scheduled for a class day will be released **one day earlier** so that you have the flexibility to pace your learning around what best suits you.

The afternoon sessions are primarily reserved for in-class exercises. In-class exercises are designed to assess students' understanding of concepts learned from video lessons and create opportunities for you to apply them into practice to solve business problems. They also provide

a venue for students to explore specific topics in greater depth and serve milestones to conclude learning outcomes on a daily basis.

As specific individual feedback is critical for making programming practice meaningful, the instructor will spend the time working with students in afternoon sessions via Zoom meeting (https://hkust.zoom.us/j/283669646). Participating in live meetings is *not mandatory*. Student can call in whenever you are in need of more explanations and clarifications on lecture contents or run into problems during working on in-class exercises.

5. Textbook and Supplemental Readings

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 includes:

- *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)

6. Lecture Videos

Due to a lack of technical support by Zoom, all lecture videos contain no subtitles. Considering possible issues concerning unclear sounds in a video, the instructor will create a discussion thread on Canvas per video so as to allow students to ask for clarification or request transcripts for unclear parts. You are highly encouraged to pause a video from time to time to try out R commands in RStudio during watching it.

7. Assessment Scheme

An inevitable part of this end of any university course is the evaluation, and the grade. Actually, in any course, the most important evaluation is a student's self-evaluation, e.g., how many new and useful ideas and skills did you learn from the course? Has the course changed your view about yourself, work groups and organizations? If so, student efforts here will have paid off.

The goals of this course will be assessed in the following manner, and the percentage of grade may be broken down as follows:

Components	Percentage of the grade
A. Labs	10%
B. In-class Exercises (9)	30%
C. Assignments (2)	30%
D. Final project (group)	30% (10% be based on your team-mate' assessment)

8. Labs

During the lab sessions, students will get hands-on practice with lecture concepts by working on assigned lab activities. So lab participation is an essential part of the learning process and thereby mandatory, and counts toward your course grade.

Lab sessions will be delivered via *live* Zoom meetings at https://hkust.zoom.us/j/140910745 at schedule times, unless an alternative mode is announced. Members of the teaching staff 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.

There will be 10 labs in total. Your Lab participation score for the course will be calculated based on the number of labs that you submit, as indicated in the table below.

9. In-class Exercises and Homework Assignments

There will be a total of **9** in-class exercises and **2** *individual* assignments, each possibly comprising conceptual questions to be answered and hands-on tasks.

Each in-class exercise is to be submitted by 11:59pm on the same date it is given. The due date of each assignment will be announced upon its release on Canvas.

10. Final Project

No final 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: -0.5% 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 1% 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 student 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.

11. Software

- On-Premises:
  - RStudio: download and install from https://rstudio.com/products/rstudio/download/ or install from within Anaconda

- Cloud:
  - RStudio Cloud
  - Google Colaboratory

12. Tentative Lecture Schedule

The following table shows the planned list of topics that we plan to cover. Please note that this schedule is tentative and is subject to adjust as the term progresses.

Date	Morning Session	Afternoon Session	Assignment Due/Remark
3-Jan	Unit 1: Course Introduction • Introduction to R and data science • Overview of planned topics • Course mechanics • RStudio • RMarkdown	Unit 2: Data Structures • R basics • Vectors and factors • **In-class exercise 1**	
6-Jan	Unit 2: Data Structures • Matrices and arrays • Lists	Unit 2: Data Structures • Data frames • **In-class exercise 2**	Add/Drop deadline
8-Jan	Unit 3: Control Structures • Conditionals • Loops	Unit 4: Functions • Writing and calling functions • **In-class exercise 3**	
10-Jan	Unit 4: Functions • Environments and scoping rules	Unit 5: Loop Functions • The apply family • **In-class exercise 4**	Asg. 1 Release
13-Jan	Unit 5: Loop functions • The split/apply/combine pattern for data analysis Unit 6: Data Wrangling • Tidy data, tidyverse, and tibbles	Unit 6: Data Wrangling • tidyr for data tidying • **In-class exercise 5**	
15-Jan	Unit 6: Data Wrangling • dplyr for data manipulation Unit 7: Base Plotting • The generic plotting function • The painter model	Unit 7: Base Plotting • Other high-level plotting functions • **In-class exercise 6**	
17-Jan	Unit 8: ggplot2 Plotting System • Grammar of graphics • Data, geoms, and aesthetic mapping	Unit 8: ggplot2 Plotting System • Other grammatical elements • **In-class exercise 7**	Asg. 1 Due Asg. 2 Release

20-Jan	Unit 9: String Operations • stringr for string operations • Regular expressions	Unit 9: String Operations • Working with tidyr and dplyr • **In-class exercise 8**	
22-Jan	Unit 10: Web Scraping • HTML and CSS basics • rvest for extracting tagged data	Unit 10: Web Scraping • Dynamic web scraping • **In-class exercise 9**	
24-Jan	Unit 11: Text Analytics • Text analytics with tidy text format • Sentiment analysis	Unit 11: Text Analytics • Tokenizing and n-grams Final project requirements Course conclusion	Asg. 2 Due

13. Tentative Lab Schedule

Lab	Date	Lab Topic
1	3-Jan	R markdown, data type, operators, vectors, and factors
2	6-Jan	Exploring data frames
3	8-Jan	Using control structures; Writing and calling functions
4	10-Jan	Practicing advanced looping and split-apply-combine manipulation
5	13-Jan	Data Wrangling with dplyr and tidyr I
6	15-Jan	Data Wrangling with dplyr and tidyr II
7	17-Jan	Plotting with R base graphics system, ggplot2
8	20-Jan	Splitting and querying with regexes
9	22-Jan	Web scraping
10	24-Jan	Text analytics and sentimental analysis