RUTGERS
Rutgers Business School
Newark and New Brunswick

Computer Science
COURSE NUMBER: 22:198:660
COURSE TITLE: Business Analytics Programming

COURSE DESCRIPTION

Our goal in this course is to learn the principles of programming for business analytics using the Python and R programming languages. Programming is the fundamental background skill based on which all Information Systems are built. Even if it is not your goal to become a software developer, it is essential for an MBA graduate with concentration in Analytics and Information Management to possess a working knowledge of programming and fundamental insights into what a programmer does. This course provides you with this essential knowledge.

COURSE MATERIALS

Recommended Textbooks:

1. Software for Data Analysis by John Chambers, Springer 2008 (PDF Downloadable from Rutgers Library).

2. Downey, Allen, Jeffrey Elkner, and Chris Meyers. *How to Think Like a Computer Scientist: Learning with Python*. Green Tea Press, 2002. ISBN: 9780971677500.
(This textbook offers an interesting approach with business and science applications)

Reference Textbooks:

ggplot2: Elegant Graphics for Data Analysis, Hadley Wickam, Springer, 2009.

R Programming for Bioinformatics by R. Gentleman, Chapman and Hall, 2008.

Learning Python, Marj Lutz, O'Reilly.

Parallel R, O'Reilly.

Advanced R Development (forthcoming) by Hadley Wickam. See Advanced R Wiki

Visualizing Data. Ben Fry. O'Reilly

CLASS ORGANIZATION & ADMINISTRATION

Prerequisites:

No previous knowledge of programming languages is required. However those of you that are familiar with some other language, particularly C or a C derivative, will have an easier ride in the first few weeks. You need to have access to a personal computer (Windows, Mac or Unix will all work.) You need to be able to download and install software on this machine. You also need to have access to the internet.

Attention:

This course is fundamentally **different** from other courses you have ever taken or will take in this program. It is **not** about learning a few formulas, principles, definitions, and applying them using the inventory of skills you have already acquired in your previous education. This course is about expanding exactly this inventory of skills that forms the underlying basis of your education to a totally **new area**, and develop a way of thinking that is *unlike* those you are employing in other coursework. Programming is not easy for those who have no prior experience with it, yet it becomes easy as you practice. Programming projects and homework are the heart and soul of this course. You have to do them in order to learn. Therefore, you may very well need to spend more time working on this course than on any other, practicing how to write programs. This is the only way you can acquire a skill essentially different from others that you already have.

Course Content:
This course presents computing tools and concepts for all stages of dealing with the modern data deluge---statistical computing at the center, but also the essential surrounding tasks, including data organization, presentation of results and the user interface. This approach is needed to deal with the challenges posed by modern technology, challenges that are also opportunities for better use of data. The size and complexity of data sources has increased enormously, while the importance of learning from the data has been recognized as never before. New modes of computing such as large-scale parallelism and cloud computing can help, but require new approaches to programming. But the key challenge is to use our own time effectively by choosing the best programming approach for each stage of a project.

To meet these challenges, we present a range of computing paradigms and corresponding languages, each designed for ease of use but also providing a rich set of tools. We use the R language and the thousands of packages written for it for core statistical computing.

Other languages are discussed for tasks where they excel. For example, Python provides a similarly strong language and a set of supporting packages for data processing, scientific computing and interactive interfaces. An approach through inter-system interfaces and interactive front ends allows you to add features from these languages without mastering all the details. Alternatively, a solution can be programmed largely in another system when appropriate, and then made available in R. Object-oriented programming techniques are particularly valuable. We discuss these both in the functional form found in R and also in the encapsulated form typical of languages such as Java and C++.

Other languages are discussed for tasks where they excel. An approach through inter-system interfaces and interactive front ends allows you to add features from these languages without mastering all the details.

Attendance:
Regular attendance is compulsory. You are not allowed to check your emails, access Web sites not related to the course or work on something that is beyond the scope of this course during the class time.

Assignments:
You may have discussions with your class members, but you have to submit your own work. Please be sure to keep a copy of the assignment by yourself in case that there is any problem with your hand-in/online submission or you have to use it later this semester. Assignments have to be submitted **before** the beginning of the class on the specified due day. **No late submissions will be accepted.**

Exams: There will be **no make-up exams.** You are required to present a written proof for situations such as going on to an emergency room due to unexpected and serious illness. Chatting during the exam is **not** allowed. **Email communication during the exam will be considered cheating.** **No** collaboration between class members will be allowed during any exam. There will be **no** extra-credit project.

FINAL GRADE ASSIGMENT

In-class work, Assignments	30%
Exam I	20%
Exam II	20%
Final	30%

Computer Science **(22:198:660)**

COURSE SCHEDULE

	Week	**Topic**	**Assignments/Quiz**
01/21	1	Introduction to BAP Functional programming and R; objects in R	(HW0 assigned – not graded)
01/28	2	Dataframes in R R packages Design, checks, publishing	
02/04	3	S4 Classes and Methods.	(HW1 assigned)

02/11	4	OOP computing model in R, Reference Classes and other languages	
02/18	5	Databases, SQL, ODBC, drivers and interfaces from R (DBI) XML, Xschema, XSL	
02/25	6	Intersystem interfaces: R and C,C++, Python, Java, etc Spreadsheet model of Computing, interface to R	(HW2 assigned, HW1 due)
03/04	7	Exam I	
03/11	8	Data Visualization: R graphics, ggplot2, graphs Python Types and Operations, Statements and Syntax	
03/18	9	SPRING BREAK	
03/25	10	Debugging R, interactively in R and at the C level Python Functions and Generators Python Modules and Packages	(HW3 assigned, HW2 due)
04/01	11	Python Classes and OOP, Exceptions and Tools Large computations and large data; vectorizing; measuring efficiency Cluster Computing, MPI, R	

		facilities, CUDA examples if time permits	
04/08	12	Cluster Computing, MPI, R facilities, CUDA examples if time permits Working with Python and R	(HW4 assigned, HW3 due)
04/15	13	Exam II	
04/22	14	Map-reduce computations, Hadoop Integrating R with Python, and Java APIs	
04/29	15	Web based interfaces, libraries, publishing. Examples Review	(HW4 due)
05/08	16	Final Exam Period 05/08 to 05/14	

Collaboration and Cheating: Collaboration of any kind is **strictly forbidden** on all exams, and quizzes. Any violations that I detect will be formally prosecuted. Students should familiarize themselves with the RBS honor code pledge, "I pledge, on my honor, that I have neither received nor given any unauthorized assistance on this examination (assignment)." See http://academicintegrity.rutgers.edu/academic-integrity-at-rutgers for more information.

Scholastic Dishonesty Policy: The University defines academic dishonesty as cheating, plagiarism, unauthorized collaboration, falsifying academic records, and any act designed to avoid participating honestly in the learning process. Scholastic dishonesty also includes, but not limited to, providing false or misleading information to receive a postponement or an extension on assignments, and submission of essentially the same written assignment for two different courses without the permission of faculty members. The purpose of assignments is to provide individual feedback as well to get you thinking. Interaction for the purpose of understanding a problem is not considered cheating and will be encouraged. However, the actual solution to problems *must* be one's own.