R Programming

Roger D. Peng

Course Description

In this course you will learn how to program in R and how to use R for effective data analysis. You will learn how to install and configure software necessary for a statistical programming environment, discuss generic programming language concepts as they are implemented in a high-level statistical language. The course covers practical issues in statistical computing which includes programming in R, reading data into R, accessing R packages, writing R functions, debugging, and organizing and commenting R code. Topics in statistical data analysis and optimization will provide working examples.

Course Content

- Week 1: Overview of R, R data types and objects, reading and writing data
- Week 2: Control structures, functions, scoping rules, dates and times
- Week 3: Loop functions, debugging tools
- Week 4: Simulation, code profiling

Background lectures

Background lectures about the content of the course with respect to other quantitative courses, course logistics, and the R programming language are provided as reference material. It is not necessary to watch the videos to complete the course, however you may find them useful.

Course Textbook

The book *R Programming for Data Science* covers all of the lecture material in this course.

Assessments

Quizzes

- There are four weekly quizzes.
- You must earn a grade of at least 80% to pass a quiz.
- You may attempt each quiz up to 3 times in 8 hours.
- The score from your most successful attempt will count toward your final grade.

Programming Assignments

- There are **three required** programming assignments.
- You must earn a grade of at least 80% to pass a programming assignment

- Programming Assignments 1 and 3 are graded via unit tests using a submission script that compares the output of your functions to the correct output.
- Programming Assignment 2 is submitted differently and graded via a peer review.

swirl Programming Assignment (practice)

- In this course, you have the option to use the swirl R package to practice some of the concepts we cover in lectures.
- While these lessons will give you valuable practice and you are encouraged to complete as many as possible, please note that they are **completely optional** and you can get full marks in the class without completing them.

Grading Policy

You must score at least 80% on all required assignments (Quizzes & Programming Assignments) to pass the course. Your final grade will be calculated as follows:

- Week 1 Quiz - 20%
- Week 2 Quiz - 10%
- Week 3 Quiz - 5%
- Week 4 Quiz - 10%
- Programming Assignment 1 (Air Pollution) - 20%
- Programming Assignment 2 (Lexical Scoping) - 10%
- Programming Assignment 3 (Hospital Quality) - 25%
- swirl Programming Assignment (practice) - 0%

Anonymity

As part of this class you will be required to set up a GitHub account. Github is a tool for collaborative code sharing and editing. During this course and other courses in the track you will be submitting links to files you publicly place in your Github account as part of peer evaluation. If you are concerned about preserving your anonymity you should set up an anonymous Github account and be careful not to include any information you do not want made available to peer evaluators.

Typos

- We are prone to a typo or two - please report them and we will try to update the notes accordingly.
- In some cases, the videos may still contain typos that have been fixed in the lecture notes. The lecture notes represent the most up-to-date version of the course material.

Differences of opinion

Please refrain from angry, sarcastic, or abusive comments on the message boards. Our goal is to create a supportive community that helps the learning of all students, from the most advanced to those who are just seeing this material for the first time.

Plagiarism

Johns Hopkins University defines plagiarism as "...taking for one's own use the words, ideas, concepts or data of another without proper attribution. Plagiarism includes both direct use or paraphrasing of the words, thoughts, or concepts of another without proper attribution." We take plagiarism very seriously, as does Johns Hopkins University.

We recognize that many students may not have a clear understanding of what plagiarism is or why it is wrong. Please see the following guide for more information on plagiarism:

JHU Student Handbook on Referencing

It is critically important that you give people/sources credit when you use their words or ideas. If you do not give proper credit -- particularly when quoting directly from a source -- you violate the trust of your fellow students.

The Coursera Honor code includes an explicit statement about plagiarism:

I will register for only one account. My answers to homework, quizzes and exams will be my own work (except for assignments that explicitly permit collaboration). I will not make solutions to homework, quizzes or exams available to anyone else. This includes both solutions written by me, as well as any official solutions provided by the course staff. I will not engage in any other activities that will dishonestly improve my results or dishonestly improve/hurt the results of others.

Reporting plagiarism on course projects

One of the criteria in the project rubric focuses on plagiarism. Keep in mind that some components of the projects will be very similar across terms and so answers that appear similar may be honest coincidences. However, we would appreciate if you do a basic check for obvious plagiarism and report it during your peer assessment phase.

It is currently very difficult to prove or disprove a charge of plagiarism in the MOOC peer assessment setting. We are not in a position to evaluate whether or not a submission actually constitutes plagiarism, and we will not be able to entertain appeals or to alter any grades that have been assigned through the peer evaluation system.

But if you take the time to report suspected plagiarism, this will help us to understand the extent of the problem and work with Coursera to address critical issues with the current system.

Getting Started and R Nuts and Bolts

This week is all about getting started with R and learning some of the basic details of the language. If you haven't already installed R, you should go to the R web site and download R for your platform (Windows, Mac, or Unix/Linux). Also, if you want, you can download RStudio, which is a free interactive development environment designed for R that is very useful and we use quite a bit in the Data Science Specialization. I've made some videos to help you along with the installation process:

- Installing R on Windows
- Installing R on a Mac
- Installing R on RStudio (on a Mac)

Before you start using R, one key concept is the **working directory**. This is the directory/folder on your computer where you will store project files, data, and code. It's important that you tell R where the working directory is that you will be using so that it knows where to find the appropriate file (the working directory can be any directory on your computer). These videos tell you how to set your working directory:

- Setting your working directory (Windows)
- Setting your working directory (Mac)

Learning Objectives

By the end of week 1 you should be able to:

- Install the R and RStudio software packages
- Download and install the swirl package for R
- Describe the history of the S and R programming lectures
- Describe the differences between atomic data types
- Execute basic arithmetic operations
- Subset R objects using the "[", "[[", and "$" operators and logical vectors
- Describe the explicit coercion feature of R
- Remove missing (NA) values from a vector

Assessments

- Quiz 1 - 80% or better required to pass
- There is **no official graded programming assignment for this week**. However, we have developed a series of practice exercises to get you started with R. These exercises are implemented using the swirl package for R. **The swirl programming assignment is NOT required**.

Practical R Exercises in swirl Part 1

During this course we'll be using the swirl software package for R in order to illustrate some key concepts. The swirl package turns the R console into an interactive learning environment. Using swirl will also give you the opportunity to be completely immersed in an authentic R programming environment. In this programming assignment, you'll have the opportunity to practice some key concepts from this course.

0. Install R

- swirl requires R 3.0.2 or later. If you have an older version of R, please update before going any further. If you're not sure what version of R you have, type R.version.string at the R prompt. You can download the latest version of R from https://www.r-project.org/.
- Optional but **highly recommended**: Install RStudio. You can download the latest version of RStudio at https://www.rstudio.com/products/rstudio/.

1. Install swirl

Since swirl is an R package, you can easily install it by entering a single command from the R console:

```
1
install.packages("swirl")
```

If you've installed swirl in the past make sure you have version 2.2.21 or later. You can check this with:

```
1
packageVersion("swirl")
```

2. Load swirl

Every time you want to use swirl, you need to first load the package. From the R console:

```
1
library(swirl)
```

3. Install the R Progroamming course

swirl offers a variety of interactive courses, but for our purposes, you want the one called R Programming. Type the following from the R prompt to install this course:

```
1
install_from_swirl("R Programming")
```

4. Start swirl and complete the lessons

Type the following from the R console to start swirl:

```
1
swirl()
```

Then, follow the menus and select the R Programming course when given the option. For the first part of this course you should complete the following lessons:

1. Basic Building Blocks
2. Workspace and Files
3. Sequences of Numbers
4. Vectors
5. Missing Values
6. Subsetting Vectors
7. Matrices and Data Frames

If you need help...

Visit the Frequently Asked Questions (FAQ) page to see if you can answer your own question immediately.

Search the Discussion Forums this course.

If you still can't find an answer to your question, then create a new thread under the swirl Programming Assignment sub-forum and provide the following information:

- A descriptive title
- Any input/output from the console (copy & paste) or a screenshot
- The output from sessionInfo()

Good luck and have fun!

For more information on swirl, visit swirlstats.com

Practice Programming Assignment: swirl Lesson 1: Basic Building Blocks

You have not submitted. You must earn 1/1 points to pass.
i

It looks like this is your first programming assignment.

Deadline Pass this assignment by Sep 28, 12:59 PM +06

1.

Complete this lesson as described in the "Practical R Exercises in swirl" reading. At the end of the lesson you will be asked to enter the email address you use to log in to Coursera and the assignment key. After entering both you will receive credit for completing this assignment.

How to submit

Copy the token below and run the submission script included in the assignment download. When prompted, use your email address **Ahmed.mesbahuddin.md@outlook.com**.

jGCmlS3Fd2GAyblB

CmuLXe9GlwNwLETy

JBDIuyRgDkKl0Wwf

jGCmlS3Fd2GAyblB

Your submission token is unique to you and should not be shared with anyone. You may submit as many times as you like.