Debugging and Profiling

10 questions

1
point

1.
Debugging is the process of

○ writing code that uses memory efficiently

● getting your expectations about code behavior to converge with reality

○ writing code that is as fast as possible

○ handling errors in a nonstandard manner

1
point

2.
Which of the following functions initiates an interactive debugging environment that allows you to step through code one expression at a time?

○ try()

○ tryCatch()

○ stop()

● browser()

1
point

3.
Which of the following functions allows you to temporarily insert pieces of code into other functions to modify their behavior?

● trace()

○ debug()

○ browser()

○ recover()

1
point

4.
What does the traceback() function do?

○ pauses execution of a function

○ evaluates an expression and returns on object of "try-error" if an error occurs

○ sets the debugging flag for a function

● returns the state of the function call stack just before an occurred

1
point

5.
When should the traceback() function be called?

○ only while debugging inside a function

● immediately after an error occurs

○ immediately after a warning occurs.

○ any time after an error occurs

1
point

6.
What does calling trace("f") for function "f()" do?

○ Whenever f() is called, the function call stack is written to a file.

● Whenever f() is called, a message is printed to the console indicating that the function was called.

○ It generates a log file containing debugging information.

○ Whenever f() is called, it will enter the interactive debugger.

1
point

7.
What is the microbenchmark package useful for?

● comparing the speed of several functions that do the same thing

○ modifying code in other developers' packages

○ handling warnings and errors in non-standard ways

○ interactively debugging functions

1
point

8.
What does the RProf() function do?

○ re-formats your code to adhere to standards

○ analyzes your code for potential bugs

● writes out the function call stack at user-specified intervals of time

○ visualizes the time behavior of R code

1
point

9.
What does the profvis() function do?

○ prints debugging information to the console

○ visualizes code that runs quickly (in less than 10 milliseconds)

● displays profiling information in an interactive visualization in RStudio

○ writes a log file of the function call stack

1
point

10.
Which function is better for analyzing fast-running functions: profvis() or microbenchmark()?

○ microbenchmark()

● profvis()

☑ I, **Vicken Asadourian**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code

Submit Quiz