Debugging

GGerardi

15.01.2023

Debugging Vs Troubleshooting

They have common traits but different processes: **troubleshooting** is the process that allows programmers to identify issues occurring within a given system while **debugging** could be seen as a subset for troubleshooting and it requires the developer to find and fix the issues that are related to computer software.

Troubleshooting is the process of extracting the elements causing the problem from a set of potential causes. This process often requires going beyond technical information and communicating with systems end users to understand what steps they need to take to identify what part of the code, infrastructure, or configuration is causing the problems.

Although **debugging** may seem simpler-finding what caused the problem within the application and then fixing it-there can be several points of failure and it is not always obvious where the problems occur. For example, a form fails not because we entered incorrect values but because the API is calling an external service that the firewall prevents from reaching. Debugging, by definition, is meant to find and fix the problem all in one shot ... even if the reality could be different.

Staying general, there are several types of errors that can occur during code writing or programming, but mainly all these errors fall into these 3 categories:

1. Syntax errors - mistakes in the source code - probably the easiest to debug and fix (modern IDE does it for us).
2. Runtime errors (exceptions) - the programming language doesn't understand what you are asking (e.g. dividing by zero)
3. Logic errors - the code doesn't operate as intended - it's a mistake in reasoning by the programmer, but it is not a mistake in the programming language.

The general rules for minimizing errors are:

1. Write comments and documentation-the most important of all;
2. DO NOT hard-code where it can be avoided;
3. Write readable code: nomenclature and typing are very important in code;
4. Make code efficient: structure code appropriately, using methods, functions and helper classes;
5. Write test cases, as specific as possible for each processing element.

Debugging in R

In R there are several tricks you can use to make finding errors easier:

1. use the **tryCatchLog** package with advanced tryCatch() and try() functions for better error handling;

2. use a logging package (e.g. **futile.logger**) to print errors to an external file and analyze the results of the code;
3. use Hadley Wickham's advice:
    - use `traceback()` which shows you the sequence of calls that lead to the error;
    - use `browser()` to enter the interactive debugger (in RStudio just use "Rerun with Debug" tool);
4. create your functions following best practices