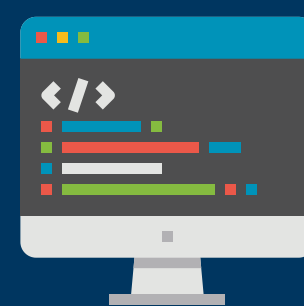# Taming the Dependency HELL

Everybody has a "dependency hell" horror story to tell. In the spookiest month of the year, we describe the leading causes of this problem and how it impacts scientific reproducibility.

The Problem

Dependency hell is a colloquial term that represents a common problem in the context of code reproducibility. It occurs when a researcher tries to run a piece of code, but the code relies on external libraries and packages or a particular version of software. These dependencies may have their own dependencies, creating a complex web of interconnected software components.

Over time, libraries and packages and the software application may have new versions or dependencies or even become obsolete and no longer actively maintained, causing scripts to break and preventing the code from reproducing results.

General Tips

- Document your computing environment (i.e., operating system, software, and versions)
- Ensure to record all packages and libraries you have used and their versions.
- Use dependency management systems and virtual environments to automate and streamline the steps above.
- Consider using containers to capture dependencies and offer an environment runtime.

Now

Scripts   Libraries/Packages   Expected Outputs

Future

Scripts   Libraries/Packages   Errors and Warnings

Deprecation
New Versions

A Scary Scenario

Access the link or scan the QR code below to listen to Dr. Stevens' experience dealing with dependencies in her research.

LET'S GET SPOOKY

tinyurl.com/dependencyhell (audio)

tinyurl.com/read-dependencyhell (text)

Stay tuned!

Upcoming issues for this series will expand on possible ways to mitigate the dependency hell problem when working with different programming languages.

UC SANTA BARBARA
Library