Module 3.3: Learn - Professional Development

3.3 Professional Development

In this installment of professional development, we will talk more about Googling errors now that you have actually started coding with real data. Learning how to identify issues in code and troubleshoot your way out of them is an essential skill for any computational scientist which can prevent you from needlessly wasting your most valuable resources – your time, energy, and patience.

Tips on solving coding errors

Debugging code

Getting errors is a natural part of the coding process and anyone who has coded knows how frustrating it can be to figure out problems. This can happen when you write code yourself and when you try to use code that other people have written. In R, you will run your code and errors will show up either in the Console or in the code output panel of the Rmd below the code chunk you are running.

Here are a list of simple things you can do to figure what is happening incorrectly in your code:

1. Read the error message R gives you very carefully
    - Well written error messages can contain the line number of the error, the type of error that took place, which variables are being set incorrectly, the data type that is not being set correctly, and many other clues that can help you know what to do next.
2. Checking the value of the variables

- Either by looking in the Environment tab in RStudio or using print statements, checking the value of va through the lines of code can help you to see where the value is not what you expect and fix whatever problem.

3. Commenting lines of code
    - By adding a hashtag (#) at the beginning of lines of code or by highlighting and using the Code menu \ Comment/Uncomment Lines option, you can omit certain lines of code that you suspect are causing a comment a line and your error goes away or changes in a way that makes sense, you may have found original error.
4. Making simple test cases
    - If you are having trouble running code using a large input file, sometimes it is easier to understand the use a subset of the file (make a copy of only the first few lines of the file) or a simpler example you cre testing. Once you get a simpler test case working, you can build out the full data set.
5. Asking others for help
    - A good first step is to ask for help from others in your class or lab or office that might have dealt with y people in the same environment are more familiar with tools you are working with and can be quicker t specific advice. If no one around you knows how to help, you can email the contact person for the pac with or search for answers from other programmers on the internet (next section).

Using internet searches to correct errors

It is very likely that if you are experiencing a problem while coding in this class, there is someone else out there w troubleshoot the same exact issue. R and many other commonly used programming languages try to include des error messages, but there are times where you don't know what the errors mean – especially when you are just g simple way to get help is simply by searching for the error message.

For example, if you enter this code into the Console in RStudio,

```
genes = ["MAPK", "EGFR"]
```

you will get an error that says:

```
Error: unexpected '[' in "genes = ["
```

If you don't know what this error means, you can get ideas by searching for the part of the error message that is r
code and indicating the programming language you are using; try:

```
"Error: unexpected '['" R
```

This time you will want to use quotes so that you get as close to your specific issue as possible. You will see a Yo
page that seems to discuss this as the top results:

**https://statisticsglobe.com/error-unexpected-comma-equal-sign-parentheses-in-r (https://statisticsglobe
comma-equal-sign-parentheses-in-r)**

If you open this page, you will see a section called "Fix the Errors" which shows a list of numbers entered using tl
of square braces, which hopefully will remind you that this is how you make a list in R. You can be iterative with y
them more broad or more specific as needed, testing out new solutions as you go.

Additional Resources

- Article: **KnightLab, Northwestern (https://knightlab.northwestern.edu/2014/03/13/googling-for-code-solutior
how-to-get-started/)** : "Googling code can be tricky…"
- Video on fixing errors/bugs in code
  - **TIPS and TRICKS on how to find and fix ERRORS, BUGS, and EXCEPTIONS in your code
(https://www.youtube.com/watch?v=9a8Y2Sx846Q)**

(https://www.youtube.com/watch?v=9a8Y2Sx846Q)

- **TIPS and TRICKS on how to find and fix ERRORS transcript (https://docs.google.com/document/d/1JeH2QojI52oR5jfqQrHcZ-iK4Wnv8rcHEnr38/edit?usp=sharing)**