Components Present in Each Section

Date: 2022-05-20

Author: Cole Brookson

Round 1

These items would be the first things created for the content in each section - the sort of "bare bones" version so to speak. This would include the secions: * *Getting started* * *Basic R* * *Working with Data* * *Plotting*

1. Written content

This is the written descriptions of all the subjects covered in each section

2. Code snippets

Example code (copyable) that illustrates the concept at hand - this is not an exhaustive program but pieces that show an output that are examples of how something is performed

3. Images of output/objects

Essential pictures to communicate where the students should be directing their attention (i.e. the environment, the console, etc.)

4. Summary & "Understanding Check" Sections

In each subsection there will be "Understanding Checks" that are optional for the students to see if they're on the right path or not - it will be a short problem or a question they can click on to see the answer to. Each overall section will have a summary that reviews the essential material.

Round 2

These items would only be created *after* the initial content is created and can be used and also receive feedback from students. This would include additions to the previously fleshed out sections as well as building out the following sections: * *Statistics* * *Programming Concepts* * *Simulations*

1. More Complete Referencing

The first round will likely be incomplete in terms of its capacity to be self-referential with regard to hotlinks etc. This is an important component of usability and so will need to be improved.

2. Extra Information Boxes

To make the content more useful to more advanced students, "further" information boxes could be added at strategic points throughout the resource that expand on concepts embedded in the main text and either illustrate a self-contained further concept, or introduce it and link to additional resources.

3. Common Pitfalls/Errors

These will demonstrate to students the most common ways that mistakes can be made in each section. They might have a component that would allow for the student to guess-and-check the solution to the broken code.

4. Downloadable Practice Code

These could be *starter* files or something more along the lines of practice code that they can play around with to get a feel for things.

5. Practice Questions

Practice questions would only be helpful if courses were directing their students to the resource and subsequently using it in such a way that students are incentivized to at least make occasional use of the practice questions.