Week 01 MPTC Notes

Shannon Tozier

Week 01 Notes

I called this qmd file "Week 01 MPTC Notes" because it contains my class notes from the first week of the MPTC course. I plan to organize this class by having a separate folder for each week (with potential subfolders in each week to separate assignments), thus this file is inside a folder called "Week_01" which is inside a folder called "MPTC". The R project is inside the "MPTC" folder. "MPTC" is inside a folder called "Duke" which has folders for each of my graduate courses. The "Duke" folder is inside the "Documents" section of my computer, because that is where I put everything.

Notes

- Rendering
  - Pieces for rendering
    - Data, text, markup, code
    - Most of these pieces are in plain-text in their own files
  - An engine assembles the pieces into a file (pdf, html, etc.)
  - We can reliably produce and reproduce the document in various formats
- Revolutions in Computing
  - Everyday computing now
    - Touch-based
    - Foregrounds a single application
    - Dislikes multi-tasking
      - Multi-tasking does not mean jumping from one thing to another

            - It means drawing together multiple pieces and putting them into a single output
        - Hides the file and operating system
        - BUT underneath it is the 1970s Unix/Command line
    - Technical Computing
        - Windows, pointer, keyboard
        - Multi-tasking via multiple windows at once
        - Exposes and leverages the file system
        - Common to use several specialized applications at the same time
        - See and interact with the 1970s Unix/Command line
        - Grounded in a paradigm increasingly far away from the everyday use of our most common computing devices
- Control, not Productivity
    - We need to confidently know and clearly show what it was that we did
    - In service of others, but mostly my future self
- Office vs Engineering Approaches
    - Each approach generates solutions to its own problems
    - Office Model
        - Like Microsoft suite
        - Formatted documents are real
        - Outputs (tables, graphs) are cut and pasted into the document
        - Changes are tracked inside the document
        - Final output is often the same format you were working in
        - Documents look like documents
        - Difficult to track where figures or results came from
        - Changes often tracked through file name
    - Engineering model
        - Plain-text files are real

        * Outputs (tables, graphs) are produced via code inside of documents
        * Changes tracked outside of files, at the project level
        * Final output assembled programmatically and converted to desired format
        * Plain-text is highly portable
        * Easy to recreate analysis
        * Project is properly version-controlled
        * Tables and figures produced and integrated programmatically
        * Can be more difficult to do simple things

- We want to do our work reproducibly
- R-Studio
    - IDE (Integrated Development Environment)
        * A kitchen is an IDE for meals
        * Brings together a lot of pieces
            · Text editor for writing code and documents
            · Console for running code interactively
            · Terminal to talk to the operating system
            · Debugger to help find problems in your code
            · File manager to navigate your project
            · Version control interface to manage changes to your code
            · Viewer for plots, tables, and other outputs
            · Inspector to see what is in your environment
        * Not required, but makes your life easier
    - 4 windows
        * Current document (top left)
            · Quarto file, write analysis
        * Console (bottom left)
            · Type or send code here, see results
        * Objects (top right)

            - Created objects appear here, also shows history of commands
        - Files, Graphs, Help (bottom right)
            - File manager
    - Your code is what is real in your project
- Writing Documents
    - Quarto
        - Successor to R Markdown
    - Want to end up with an output file that contains all the pieces of a project
        - Quarto allows us to write all the pieces together by replacing code with its output
    - Pieces of the quarto document
        - Header section with metadata
        - Text with markdown formatting
        - Chunks of code
            - When rendered, replaced by their output
            - Can have labels or options
    - Ctrl + Alt + I creates new chunk
    - Notebooks work smoothly when
        - Document is small and self-contained
        - Making a lot of similar reports from a template
    - Notebooks get awkward when
        - Analysis has many pieces
        - Project has many authors
        - Analysis needs a lot of cleaning and prep-work