Data Quality Ideas

* Good commenting on code that describes WHY the code block is needed
  + Also contact information in code
* Documenting changes via a version history document
  + This is particularly useful when making changes to Excel documents with multiple versions.
* Read Me files for code
* Multiple viewing of results before they are sent out
* Quality Assurance reviews
* Testing of code to make sure that it is working as intended
  + And documenting those tests

Documentation

* README Files
* Comments in code
* Doc strings
* Version history files

Habits for work:

* Asking why. Why is this bug happening? Why does this change fix it? Why do you want this (when speaking to customers and stakeholders).
* Be on the lookout for simplifications. Decouple, modularize, reuse, rewrite, refactor, etc.
* Document everything.
* Keep a journal (para method works well here)
* Don’t write code that you don’t need right now.
* Be proactive about learning.
* Don’t spend time optimizing until there is an actual problem as optimized solutions can be more complicated, making the work harder to understand.
* Clarity over clever.
* Don’t repeat yourself.
* Create unit tests.
* Adhere to a fixed coding style.
* Document your code
  + Use comments inside of your code
  + Write documentation in a separate document
  + Write self-documenting code
* README files
  + What is this code? What problem does it solve?
  + How can you start working on this code? Include specific instructions to create a development environment, if any.
  + How do end-users use the software?
  + Offer related links, like more documentation, background info, etc.
  + Where and how should people ask for help?
* Writing functions
  + Extract as much as possible into functions and make those functions do *one* thing. Name them accordingly, and start with a verb. For example, a function that reads a file into a string could be named “readFileToString(String filename)”. Without reading the code in detail, people will know what it does.
* Refactor code
  + This is when you restructure your code without changing its behavior.