**Unit 13 – Software Engineering**

**How to Be a 10X Data Scientist**

* You need to be a domain expert and to understand what is most important to stakeholders
* You must talk to be people to learn about the goals of the business and the data
* Create functions more often, any time you may have repeatable code
* Centralize automation function
* Tools for deploying your model; what are the cost?

**Data Science is Software Lecture**

* Tqdm for timing actions
* Driven data cookie-cutter data science
* Notebooks are not good at doing the same thing every time
  + Refactor the code somewhere else
* Setting up virtual environments is a must
* Pip install requirements.txt (minimum packages for running a given notebook)
* Cookiecutter to create a python package
* Np has an almost equal function to check to make sure code is functioning properly (assert)
* Pytest package tells us more information on why assertions fail
* Code is written for humans (machines can read 1s and 0s)

**Constructive Code Review**

* How do we perform a code review?
* I get paid so I can spend my time coding (creatively)
* Make sure that you document your rationale when you code something
* Having a style standard makes “nitpicks” standardized and automated

**Cauldron Podcast**

* Download a cauldron reader, so everyone sees your notebook exactly the same way you do

**Getting Starting Testing**

* Good tests: automated, fast, reliable, informative, focused

**Best Practices for Debugging**

* Print statement is not ideal for debugging
* Import pdb; pdb.set\_trace()
  + Enters the debugger when you run the code
  + “n” for next line
  + “l” lets you see where you are
  + “s” steps into a function
  + “b 68” to create a break point at line 68
  + “c” to continue until you reach the end or a break point
  + “q” to quit session
* Logging is super usable in python
  + If you build a pipeline, logging can track the iterations
  + Can give timestamp