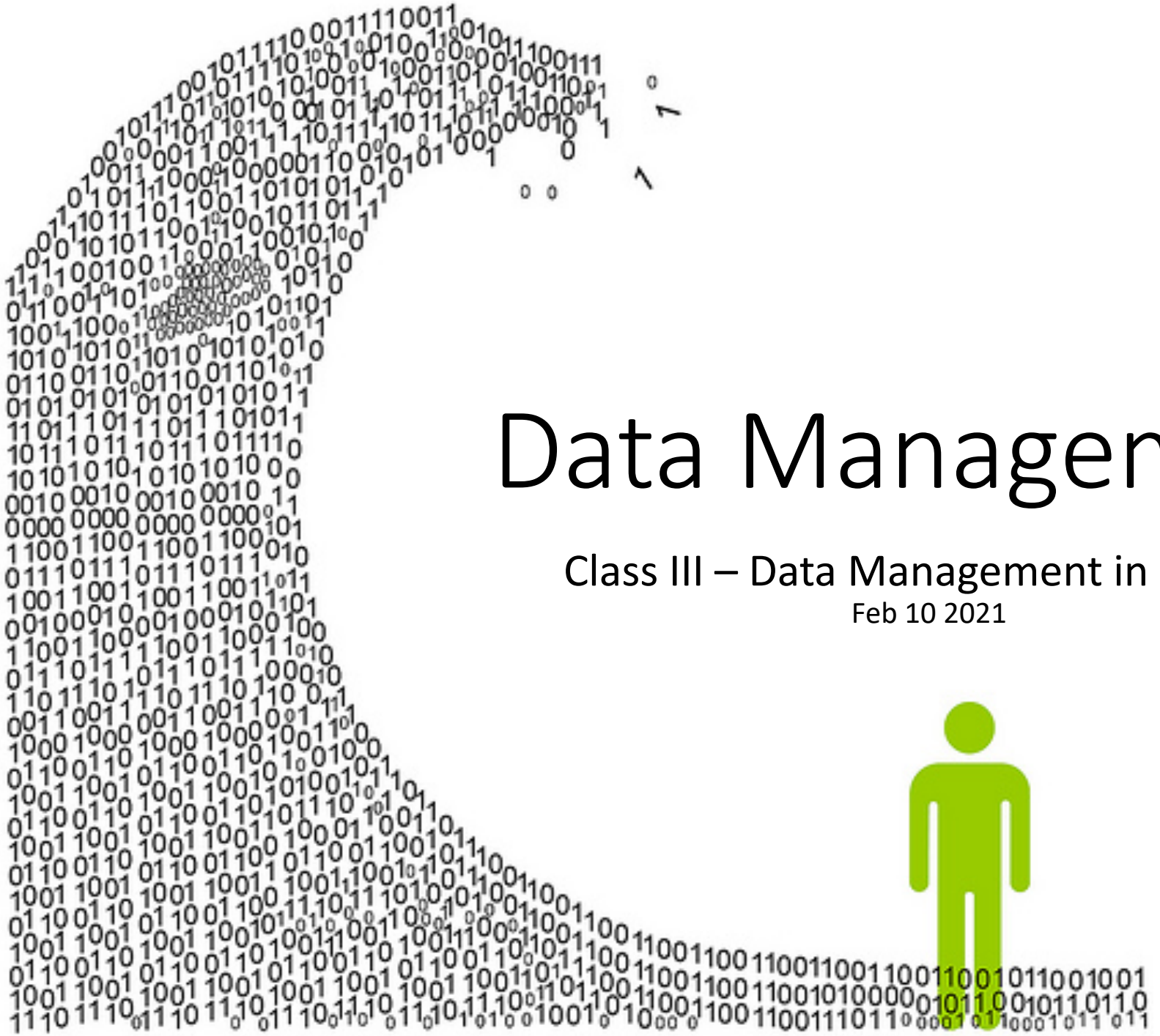
Data Management

Class III – Data Management in Python

Feb 10 2021



Recap

- In yesterday's class, we saw that there are a variety of different ways to approach data management – from something as simple as sharing data files on a USB stick, to something as complex as managing a large-scale data architecture in the Cloud.
- In future classes, we'll move on to looking at how to use common types of database – but for the next couple of days, I want to focus on the most simple (in theory!) kind of data management, **file sharing**.

File Sharing: Deceptively Simple

- In theory, file sharing really is simple. I save a file from R:
    - `save(mydata, file = "mydata.Rdata")`
- And now I can just put that file on a USB drive and give it to you. Simple!
- Well…
- Sometimes, it really is simple. This method is absolutely fine if you're working with a colleague who is also using the same R software as you.
- But what if your colleague isn't using R? Or if you need to share your data with a bigger team, or with the public?

When “Simple” isn’t really Simple

- There are several problems with RData files.
  - They only work in R, so anyone using different software won’t be able to load the data (at least not easily).
  - They’re not guaranteed to work with every version of R, so in future someone might not be able to load your file because they’re using a newer version of R.
    - You might even lose access to your own data when you upgrade your computer or your software!
  - Even if you’re using R, you need to know what’s actually in an RData file to use it effectively; just loading those data into your copy of R and looking at them often isn’t sufficient to understand what the data actually *is*.

Portability

- All of these problems can be summarised by saying that RData files are not **portable**.
- In data science, the “portability” of a type of file refers to how easy it is to move that file between different computers, operating systems, programming languages and applications.
- A file that only opens in one application, and no other application can read it, is not portable.
  - RData files are very low in portability - no other software except R can read them easily.
  - Excel files are a little more portable; although that kind of file is exclusive to Microsoft Excel, many other applications (like R!) can read data from a simple Excel spreadsheet.

Portable File Types

- Luckily, there are several types of file which were invented precisely to fix this problem – by being completely **portable**, so they are not tied to one specific language or software.
- The oldest and simplest of these are:
  - Plain text files (which usually have the file extension .TXT)
  - Comma-Separated Value files (which usually have the file extension .CSV)
- Although they're useful for many kinds of data, these can really only store simple pieces of text or tables of data.
  - More modern formats, such as "Feather", have been invented to let more complex data – like multi-dimensional frames of data – be shared easily.

Portability: R to Python (and back again)

- For data scientists, one of the most important portability challenges is making sure your data works in both R and Python.
- These are the two most popular languages in the data science world, and many research projects need to use both of them.
  - In fact, most accomplished data scientists are “bilingual” in R and Python, and it’s common to know some other languages (such as C and JavaScript) too.

Today's Objective

- In today's class, we're going to take a look at how Python handles loading data files, and do some simple analysis tasks.
- You don't need to become a Python expert – DAPS&CO is mostly taught using R, after all – but it's helpful to understand how other popular languages like Python work, so you can bear it in mind when you're designing your data management strategy.
- It's also good to get hands-on with Python - some analysis tools are only available for Python, but with experience, you'll be able to use them as easily as R tools.