Programming Environment

Over the years, we have experimented with a range of approaches to setting you up with a programming environment: VirtualBox; Vagrant; Docker; and Anaconda Python. Each of these has pros and cons, but after careful consideration we have come to the conclusion that **Docker** is the most robust way to ensure a consistent experience in which all students end up with the same versions of each library, difficult-to-diagnose hardware/OS issues are minimised, and running/recovery is the most straightforward.

Some students are **unable to run Docker** on their older Windows 10 Home machines, in which case **Anaconda Python** can be used with the configuration file that we provide. However, if your machine runs Docker then you **must** use Docker: this isolates the programming environment form your computer, ensuring that nothing is clobbered by accident, and guaranteeing that you are working with the same version of every Python library as the rest of the class (and for which the practicals are tested).

Anaconda is only for emergencies.

Docker

> Known issues
>
> On Windows, if you cannot see any of the files on your *main* system when you start up Docker with the SDS image, then please try replacing this part of the Docker command (see details below):
> `-v "$WORK_DIR":/home/jovyan/work`
> with:
> `--mount type=bind,source="$(pwd)",target=/home/jovyan/work`
> This should enable you to see any existing files that you have, while also allowing you to save any files that you create.

As well...

> Tip
>
> On Mac, if you are using one of the new M1 or M2 chips on your *main* system then you need to tell Docker to emulate the older Intel chipset (which is all I have

access to!) by changing this part of the Docker `run` command (see details below):
`--name sds -p 8888:8888`
to:
`--name sds --platform linux/amd64 -p 8888:8888`

Anaconda Python

Again, we *only* support Anaconda Python as a fallback for students who would otherwise be unable to complete the module because their computer does not support Docker. You are always free to install Anaconda Python and to use our YAML configuration script to install the SDS environment, but you should do this in your own time: in previous years students have encountered difficult-to-diagnose bugs in their code (and lost marks in the Assessments!) because they had installed an older or more recent version of a Python library that the one configured and tested in the SDS environment.

We believe that the replication advantages of *virtualisation* outweigh the disbenefits in terms of *performance*. It also means that you will spend less time installing libraries and more time running code, which is where your attention should really be when you are familiarising yourself with the *foundations* of data science. Eventually you will, of course, want to install and manage your own programming environment (possibly even by building and sharing Docker images!) but this can be left for later when you have developed an appreciation of how and when virtualisation is (or is not) an appropriate solution to your needs.