**Setup**

After logging into Rstudio server, starting a lesson requires the right version of R, the right libraries, and to start the lesson.

**R version**

This cannot be changed in the free version of Rstudio. Everyone uses the same version, which can be specified in /etc/rstudio/rserver.conf

` rsession-which-r=/usr/local/lib/R `

We are currently using R 3.6.3 (on Linux)

**Package libraries**

Both scripts require libraries. Library management on the server is difficult because the base environment must remain compatible for everyone (read: is never updated).

The solution is to create a separate ~~(conda)~~ environment, and exclusively use this. More on this in chapter 7.

**Starting a lesson (relatively) seamlessly**

Once logged into Rstudio server, a user must be able to start a lesson. Several conditions apply:

* We do not want to expose users to coding at this point (copy pasting at most)
* We do not want to allow editing of the lessons
* This needs to work without installing anything

Running the lessons from Rstudio means we must start them manually. This can be done from the lesson .Rmd file, or via a startup script. This script might be incorporated into a launcher or dashboard to improve the ease of use.

I created two types of startup scripts (I created the 2nd when I thought the first wasn’t working):

* Start a lesson as shiny app
* Start a lesson as a learnr tutorial

**Tutorial packaging**

In order make a lesson into a tutorial, it must be incorporated into an R package.

In order to set library use, or start the tutorial, we need a script.

A potential dashboard to start the tutorials in would be a separate script too.

To best suit our need we need to create 2 R packages:

* one control package to (create a dashboard,) set libraries and run tutorials,
* one tutorial package

the control package could eventually be used for other tutorials as well.