Dear PopSim nlmixr2 PK/PD course participant,

PopSim is coming closer and preparations to run the course are in full swing! If you want to take full advantage of this hands-on course it means that you need to bring a laptop with the software pre-installed. There is no time to help you with installation issues on the day of the course and so the instructions below will need to be followed to ensure your participation.

Both nlmixr2 and rxode2 are software packages that run inside R-software and so the first prerequisite is to have R software installed. If you already have R on your laptop you can add the required packages, but these packages run best with R-4.4.0 and so you might consider setting up a separate version just for the course. The instructions below apply to Windows; it is possible to run the software on mac or Linux but course code has not been checked to run smoothly on those operating systems (see hints below). If you have a (company) laptop that does not allow installation of software you will either need to contact your system administrator or bring a non-locked (personal) laptop.

If R is not yet available on your laptop or if you want a separate install, then install R-4.4.0 using the default settings with the download from <https://cran.r-project.org/bin/windows/base/R-4.4.0-win.exe>. Next, installation of the compilers from the Rtools package is required. For R-4.4.0, Rtools 4.4 is necessary, and can be obtained from from <https://cran.r-project.org/bin/windows/Rtools/>. If you are using R-4.3.0 you would need Rtools 4.3.

Rstudio is very useful as an environment to run R, and can be obtained from <https://posit.co/download/rstudio-desktop>. If you already have Rstudio installed you may want to update to the latest version.

With these prerequisites dealt with, you are ready to install the course software. Open up R, preferably in Rstudio and run the following command:

install.packages(c("nlmixr2","ggPMX","xpose.nlmixr2","babelmixr2", "nonmem2rx","xpose.nlmixr2","nlmixr2lib","patchwork"), dependencies=TRUE)

After the installation has finished (it will download a lot of packages) you can check to see if nlmixr2 is installed correctly by running:

library(nlmixr2)

nlmixr2CheckInstall()

This should allow you to run all the code on Windows during the course. Full course materials (scripts, presentations etc) can be downloaded from <https://blog.nlmixr2.org/courses/>

Installation of nlmixr2 is definitely possible on a Mac (see <https://nlmixr2.org/>) but is not supported during the course. Alternatively, you can run nlmixr2 in a so-called Docker container. Full installation instructions, both to install Docker software, and to install the Docker container with all the required software, can be found at <https://github.com/RichardHooijmaijers/nlmixr.docker>

Now the next bit sounds childish, but we’ve done too many courses to know that it is rare for all participants to come fully prepared. Therefore, we ask that you run the following script, and send the resulting objective function value to [rik.schoemaker@occams.com](mailto:rik.schoemaker@occams.com) as a prerequisite to join the hands-on part of the course.

library(nlmixr2)

one.compartment.IV.model <- function(){

  ini({

    lCl <- 1.6      #log Cl (L/hr)

    lVc <- 4.5      #log V (L)

    prop.err <- 0.3

    eta.Vc ~ 0.1

    eta.Cl ~ 0.1

  })

  model({

    Vc <- exp(lVc + eta.Vc)

    Cl <- exp(lCl + eta.Cl)

    d / dt(centr) = -(Cl / Vc) \* centr;

    cp = centr / Vc;

    cp ~ prop(prop.err)

  })

}

fit <- nlmixr2(one.compartment.IV.model, Bolus\_1CPT,  est="focei")

fit$OBJF

Full course materials can be downloaded from <https://blog.nlmixr2.org/courses/> and you can find additional information at <https://nlmixr2.org/>, <https://blog.nlmixr2.org/>, <https://opensource.nibr.com/xgxr/>, and <https://github.com/ggPMXdevelopment/ggPMX>

Looking forward to a great course!

Rik Schoemaker on behalf on the nlmixr2 team