Useful R Utilities

Jason Thomas

November 8th, 2018

In collaboration with the WHO VA Reference Group

Topics for this morning

- Packages in `R`
  - listing, updating, and installing
  - configuring `rJava`
- Saving figures
- Running `R` in *batch* mode
- Creating reports in `R` with RMarkdown

R Packages: introduction

- An R package is a collection of related functions (and potentially data) that are useful for solving a particular problem.
- All of the packages installed on your computer are stored in your R `library`.
  - List the packages already installed on your computer with the `library()` command (without any arguments)

```
> library()
```

  - `help()` provides more information about a package. Here is an example of the help file for the MASS package

```
> help(package = MASS)
```

R Packages: updating packages

- R and its packages are usually updated every few months.
  - To use the latest version of R we have to
  - There are also software managers (e.g., APT) that can handle updates for you (you will probably still have to update your library)
- To see which R *packages* have updates, we can use the `old.packages()` command.
- Updating *packages* is done with the following command:

```
> update.packages()
```

R Packages: installing packages

- Most R packages can be installed and loaded with ease.

```
> install.packages("CrossVA")
```

- On your own computer, R will ask you...
  - to choose a mirror from where you would like to download the package
  - and specify a folder where you would like to install the package (R will suggest a location)
- After installing the package we can load it (and thus have access to the functions) with as follows

```
> library(CrossVA)
```

R Packages: additional notes

- Most packages have dependencies – i.e., their functions use tools from other R packages
- `install.packages()` will also install the dependencies for you
- Some packages have A LOT of dependencies, so `install.packages()` may take a while to run (and R will produce a lot of output/messages)

Installing Packages: `rJava`

- The `rJava` package allows `R` to call (*usually faster*) Java programs
  - `R` is an extremely flexible programming language, which makes it relatively slow (among other reasons).
  - This is what the `InSilicoVA` package does.
- Configuring `rJava` may require an additional step or two (compared to other packages).
  - (RStudio is usually pretty good at figuring this out on its own.)
  - Mac & Linux: within a terminal...

```
> R CMD javareconf
```

Installing Packages: rJava on Windows

On Windows, try one of the following commands at the R prompt...

```
> ## option 1
> options(java.home = "C:\\Path\\to\\Java\\jdk")
```

```
> ## option 2
> Sys.setenv("JAVA_HOME" = "C:\\Path\\to\\Java\\jdk")
```

If these fail, try adding the environment variable JAVA_HOME set equal to the path to jdk (and restarting your computer).

Saving Figures in R

After producing a figure in R we can save the plot as a file to send to others or include in reports.

```
> x <- rnorm(100)
> pdf(file = "plotDensity.pdf") ## save as pdf
> plot(density(x))              ## create plot
> dev.off()   ## tell R we are done and to create PDF file
```

Similar commands for saving in different formats:

- `jpeg()`
- `png()`
- `wmf()` – MS Windows metafile

Running R in *batch* mode

- Sometimes it is useful to have R run all your code and create a log of the output.
- We can accomplish this by running R in *batch* mode.
- In RStudio, we can use batch mode in the Terminal (it is the tab next to the Console)...
  - change into the folder with our R (e.g., intro.R)
  - and use the following command

```
$ R CMD BATCH intro.R
```

Creating reports in R with RMarkdown

- There is a package called `rmarkdown` that is **VERY** useful for creating reports in R.
  - Reports can be pdf (required LaTeX) or MS Word Files
- There are several dependences (e.g., pandoc) but everything you need is installed in the virtual machine and on R Studio Cloud.
- R Studio (in the Cloud and on your own computer) has a useful template and great documentation for getting started.
  - https://rmarkdown.rstudio.com/
  - Quick Tour
  - Tutorial