# UBISS16 - Transversal Skills in R

> Proposal of contents for some session/s of the course UBISS16 - Data Analysis and DoE in R (advanced) (from this summer school ) By Xavier de Pedro Puente, Ph.D.

Regardless of your discipline, you usually need to achieve some transversal competences for your day to day work using R.

**Table of contents**

# 1.1. Reproducible Research (1-2h)

You can do your experiment, and get your results. but what would you do when you want to repeat the same experiment / analysis in a few years, when R version and the R packages you used have changed and might not be compatible with the parameters or conditions of your current analysis?

How do you easily tell others what your code does, so that they can reproduce it step by step?

How do you reproduce some results when you gave students a process with some random number generated in the script?

This section will teach you how to solve these type of issues, so that your research is easily reproducible.

### 1.1.1. mran repository + checkpoint package

### 1.1.2. packrat

### 1.1.3. Markdown to embed code and output

### 1.1.4. seeds in random process

### 1.1.5. Limitations and concluding remarks

---

## 1.2. Reports in html (1-2h)

You have produced your analysis results, and you want to tell the world (or your customer) about it, without requiring complicated steps (requiring specific programs that might not be available in the computer or mobile device of your readers) to view your results.

You can use html reports, so that they can be easily seen by anyone, regardless of the Operating System they use, or device (tablet, smartphone, ...), and they can be seen at any time.

### 1.2.1. custom reports via markdown + knitr

### 1.2.2. custom reports via packages (e.g. Nozzle.R1, ...)

### 1.2.3. Embed dynamic objects

Beyong static charts and graphs[1] ....

#### 1.2.3.1. DT tables

See: DT: An R interface to the DataTables library
https://rstudio.github.io/DT/[2]

### 1.2.3.2. Google Charts and tables

See: googleVis examples:
https://cran.r-project.org/web/packages/googleVis/vignettes/googleVis_examples.html[3]

### 1.2.3.3. RCharts & htmlwidgets

**rCharts** is an R package to create, customize and publish interactive javascript visualizations from R using a familiar lattice style plotting interface.
See: http://rcharts.io[4]

**htmlwidgets**: Bring the best of JavaScript data visualization to R. Use JavaScript visualization libraries at the R console, just like plots. Embed widgets in R Markdown documents and Shiny web applications. Develop new widgets using a framework that seamlessly bridges R and JavaScript
See: http://www.htmlwidgets.org[5]

## 1.2.4. Where to publish them

**RPubs** (among others). Easy web publishing from R. Write R Markdown documents in RStudio. Share them here on RPubs. (It’s free, and couldn’t be simpler!)

See https://rpubs.com[6]

---

# 1.3. Web Apps with R (1-2h)

Sometimes you need something more than just to display your results with some sort of dynamic objects embedded, but you need to create full web applications. You can do so through different strategies or tools.

## 1.3.1. Shiny apps

See: http://shiny.rstudio.com/gallery/[7]

## 1.3.2. OpenCPU

An API for Embedded Scientific Computing. OpenCPU is a system for embedded scientific computing and reproducible research. The OpenCPU server provides a reliable and interoperable HTTP API for data analysis based on R. You can either use the public servers or host your own.

See: https://www.opencpu.org[8]

## 1.3.3. Through Wiki/CMS + R:

### 1.3.3.1. Tiki + R

For full control on the application: users, groups, menus, theme style (Bootstrap ready), categories of content, email notification of changes per user or per group, searching content respecting user permissions, trackers (databases), knowledge bases (wiki), discussion forums, etc. and all integrated and free/libre open source.

See https://r.tiki.org[9]

### 1.3.3.2. MediaWiki + R

See: https://www.mediawiki.org/wiki/Extension:R[10]

### 1.3.3.3. Concerto + R

Open-source Online R-based Adaptive Testing Platform. Concerto is an open-source testing platform that allows users to create various online assessments, from simple surveys to complex IRT-based adaptive tests.

See: http://www.psychometrics.cam.ac.uk/newconcerto[11]

# 1.4. Team collaboration in R with git & github (1-2h)

For the time being, see some outline here:
http://ueb.vhir.org/SeminarDVCS[12]

---

# 1.5. Profiling + Optimization (1-2h)

Sometimes you have a powerful computer with many cpu's, but R only uses one by default in most cases. Or you are jnot taking profit of some of the extra capabilities that can be used in R for improved performance. This section will train you on alternatives to improve your code performance.

See general information here:

- https://csgillespie.github.io/efficientR/[13]

## 1.5.1. RRO version

Microsoft R Open, formerly known as Revolution R Open (RRO), is the enhanced distribution of R from Microsoft Corporation. It is a complete open source platform for statistical analysis and data science.

The current version, Microsoft R Open 3.2.4, is based on (and 100% compatible with) R-3.2.4-

Revised, the most widely used statistics software in the world, and is therefore fully compatibility with all packages, scripts and applications that work with that version of R. It includes additional capabilities for improved performance, reproducibility, as well as Windows, Mac OS X, and Linux based platform support.

https://mran.microsoft.com/open/[14]

## 1.5.2. Code Profiling

See:
"Code Profiling in R: A Review of Existing Methods and an Introduction to Package GUIProfiler"
https://journal.r-project.org/archive/2015-2/rubio-villar.pdf[15]

See reference document by Hadley Wickam:
http://adv-r.had.co.nz/Profiling.html[16]

## 1.5.3. Code Optimization tips

Speed improvements with better coding vs. Parallelized runs

See: http://www.r-bloggers.com/strategies-to-speedup-r-code/[17]

Speed Summary

Method: Speed, nrow(df)/time_taken = n rows per second
Raw: 1X, 120000/140.15 = 856.2255 rows per second (normalised to 1)
Vectorised: 738X, 120000/0.19 = 631578.9 rows per second
True Conditions only: 1002X, 120000/0.14 = 857142.9 rows per second
ifelse: 1752X, 1200000/0.78 = 1500000 rows per second
which: 8806X, 2985984/0.396 = 7540364 rows per second
Rcpp: 13476X, 1200000/0.09 = 11538462 rows per second

The numbers above are approximate and are based in arbitrary runs. The results are not calculated for data.table(), byte code compilation and parallelisation methods as they will vary on a case to case basis, depending upon how you apply it.

Code compiling:
http://www.r-statistics.com/2012/04/speed-up-your-r-code-using-a-just-in-time-jit-compiler/[18]

## 1.5.4. Code Parallelization tips

MUST READ Introduction:
http://michaeljkoontz.weebly.com/uploads/1/9/9/4/19940979/parallel.pdf[19]

Then see notes and tips at:
http://ueb.vhir.org/Parallel+R[20]

# 1.5.5. Debugging with RStudio

See reference contents from here:
https://support.rstudio.com/hc/en-us/articles/205612627-Debugging-with-RStudio[21]

Alias names for this page:
TransversalSkills | TransversalSkillsInR | Transversal Skills in R | UBISS16-TransversalSkillsinR | UBISS16 Transversal Skills in R

[1] http://www.joyce-robbins.com/wp-content/uploads/2016/04/effectivegraphsmro1.pdf

[2] https://rstudio.github.io/DT/

[3] https://cran.r-project.org/web/packages/googleVis/vignettes/googleVis_examples.html

[4] http://rcharts.io

[5] http://www.htmlwidgets.org

[6] https://rpubs.com

[7] http://shiny.rstudio.com/gallery/

[8] https://www.opencpu.org

[9] https://r.tiki.org

[10] https://www.mediawiki.org/wiki/Extension:R

[11] http://www.psychometrics.cam.ac.uk/newconcerto

[12] http://ueb.vhir.org/SeminarDVCS

[13] https://csgillespie.github.io/efficientR/

[14] https://mran.microsoft.com/open/

[15] https://journal.r-project.org/archive/2015-2/rubio-villar.pdf

[16] http://adv-r.had.co.nz/Profiling.html

[17] http://www.r-bloggers.com/strategies-to-speedup-r-code/

[18] http://www.r-statistics.com/2012/04/speed-up-your-r-code-using-a-just-in-time-jit-compiler/

[19] http://michaeljkoontz.weebly.com/uploads/1/9/9/4/19940979/parallel.pdf

[20] http://ueb.vhir.org/Parallel+R

[21] https://support.rstudio.com/hc/en-us/articles/205612627-Debugging-with-RStudio