

Controlling dependencies

Francisco Rodriguez-Sanchez

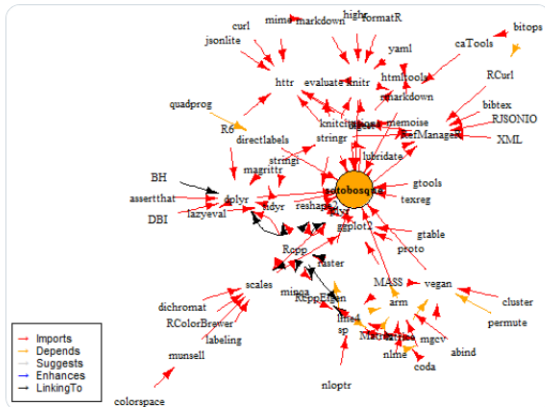
<https://frodriguezsanchez.net>



F Rodriguez-Sanchez

@frod_san

Our last project depends on complex ecosystem of 67 co-evolving [#rstats](#) pkgs. Ensuring [#reproducibility](#) not trivial



5:39 pm · 27 Jan 2016 · Twitter Web Client

Package changes can break your analysis

How to reproduce your analysis in a year,
or different computer?

sessionInfo records OS & used packages

```
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
```

```
Platform: x86_64-pc-linux-gnu
```

```
Running under: Ubuntu 20.04.6 LTS
```

```
Matrix products: default
```

```
BLAS: /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
```

```
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3; LAPACK version 3.9.0
```

```
locale:
```

```
[1] LC_CTYPE=en_GB.UTF-8      LC_NUMERIC=C
[3] LC_TIME=es_ES.UTF-8      LC_COLLATE=en_GB.UTF-8
[5] LC_MONETARY=es_ES.UTF-8  LC_MESSAGES=en_GB.UTF-8
[7] LC_PAPER=es_ES.UTF-8     LC_NAME=C
[9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
```

```
time zone: Europe/Madrid
```

```
tzcode source: system (glibc)
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] knitr_1.49
```

```
loaded via a namespace (and not attached):
```

```
[1] compiler_4.4.2    fastmap_1.2.0     cli_3.6.3         htmltools_0.5.8.1
[5] tools_4.4.2       rstudioapi_0.17.1 yaml_2.3.10       codetools_0.2-20
[9] rmarkdown_2.29    binb_0.0.7        xfun_0.50         digest_0.6.37
[13] rlang_1.1.4       evaluate_1.0.1
```

checkpoint reconstructs packages in given date

```
library('checkpoint')  
  
options(checkpoint.mranUrl="https://packagemanager.posit.co/")  
  
checkpoint('2019-10-08')  
  
source('analysis.R')
```

1. Detects packages used
2. Installs version from given date (only CRAN)
3. Independent install (not messing w/ main library)

automagic records & install packages (CRAN + GitHub)

```
automagic::make_deps_file()
```

File `deps.yaml` records dependencies:

```
- Package: equatiomatic  
  Repository: CRAN  
  Version: 0.1.0  
  
- Package: report  
  GithubUsername: easystats  
  GithubRepo: report  
  GithubRef: HEAD  
  GithubSHA1: c48a4bb0a40df7116bc502aa3ce2cbbc9d70b7e2
```

To install all those dependencies:

```
automagic()
```

renv: recommended to control dependencies

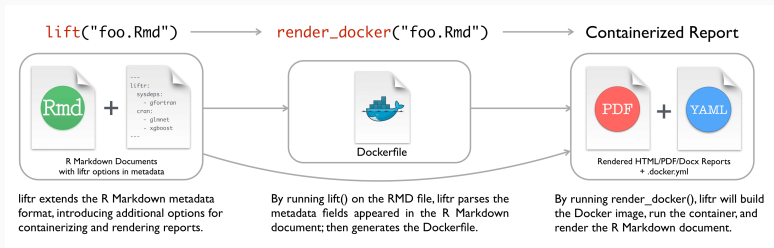
```
renv::init()  
# Create private package library for project  
  
renv::snapshot()  
# Capture dependencies in lockfile  
  
renv::restore()  
# Regenerate dependencies from lockfile
```

<https://rstudio.github.io/renv/>

To ensure reproducibility,
besides R packages
we also need to control
computational environment

Docker recreates virtual systems
from a **Dockerfile**

liftr: process Rmd in Docker container



<https://liftr.me/>

containerit creates Dockerfile

```
library('containerit')  
  
dockfile <- dockerfile(from = 'mypaper.Rmd')
```

<https://o2r.info/containerit>

tugboat created Dockerfile w/ entire software environment

tugboat



A simple R package to generate a Dockerfile and corresponding Docker image from an analysis directory. tugboat uses the [renv](#) package to automatically detect all the packages necessary to replicate your analysis and will generate a Dockerfile that contains an exact copy of your entire directory with all the packages installed.

tugboat transforms an unstructured analysis folder into a `renv.lock` file and constructs a Docker image that includes all your essential R packages based on this lockfile.

tugboat may be of use, for example, when preparing a replication package for research. With tugboat, you can take a directory on your local computer and quickly generate a Dockerfile and Docker image that contains all the code and the necessary software to reproduce your findings.

```
library(tugboat)
create()
build()
```

<https://www.dmolitor.com/tugboat/>

Your turn

- Create script/Rmd using different packages
- Call **checkpoint** on former date
- Record dependencies:
 - `renv::snapshot`
- Recreate packages
 - `restore()`