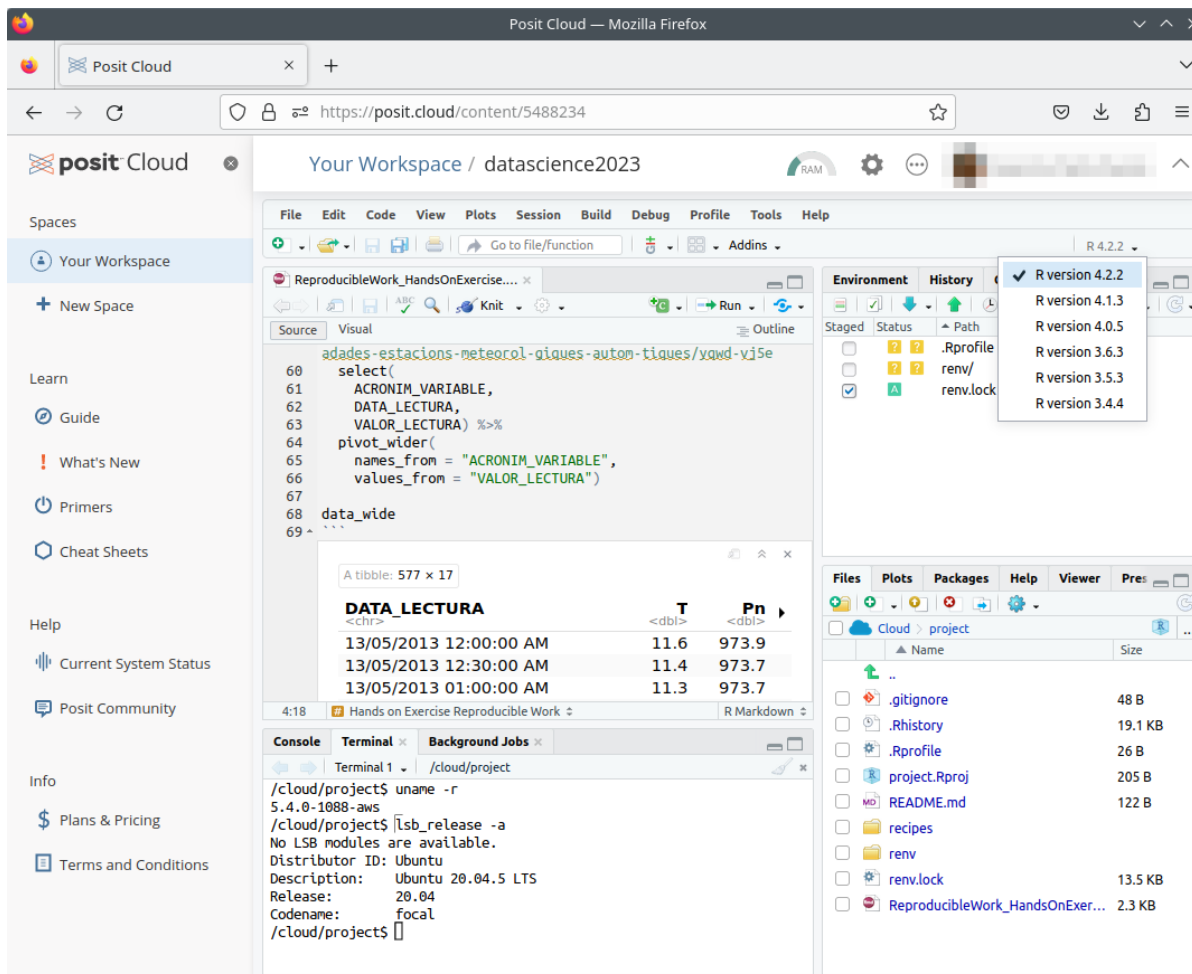


2024 Reproducible work in Data Science (X. de Pedro)

"Data Science. Applications to Biology and Medicine with Python and R", at IL3 - University of Barcelona^[1]. April 10th, 2024 (16-19:15h).

Content at <https://seeds4c.org/reproduciblework2024>^[2]

Slides in PDF



The screenshot displays the Posit Cloud interface in a Mozilla Firefox browser window. The main workspace area shows an R script with the following code:

```
adades-estacions-meteorol-giques-auton-tiques/yqwd-vi5e
select(
60   ACRONIM_VARIABLE,
61   DATA_LECTURA,
62   VALOR_LECTURA) %>%
63   pivot_wider(
64     names_from = "ACRONIM_VARIABLE",
65     values_from = "VALOR_LECTURA")
66
67 data_wide
68 ...
69
```

Below the code, a tibble is displayed with the following data:

DATA_LECTURA	T	Pn
13/05/2013 12:00:00 AM	11.6	973.9
13/05/2013 12:30:00 AM	11.4	973.7
13/05/2013 01:00:00 AM	11.3	973.7

The interface also includes a terminal window at the bottom showing the following commands and output:

```
/ccloud/project$ uname -r
5.4.0-1088-aws
/ccloud/project$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.5 LTS
Release:        20.04
Codename:       focal
/ccloud/project$
```

On the right side, there is a file explorer showing the project structure, including files like .gitignore, .Rhistory, .Rprofile, project.Rproj, README.md, recipes, renv, renv.lock, and ReproducibleWork_HandsOnExer... The environment menu is open, showing R version 4.2.2 selected.

- 2024 Reproducible work in Data Science (X. de Pedro)
- 1. Introduction - the problems (i)
 - 1.1. The problems (ii)
 - 1.2. The problems (iii)
 - 1.3. The problems (iv)
 - 1.4. The problem (v)
 - 1.5. The problem (vi)
- 2. Enemies of reproducibility & adaptability
- 3. Reproducibility & Adaptability
- 4. Reproducibility & Adaptability - Example in Posit Cloud

- 4.1. Level 1: Virtual Machines or Containers
- 4.2. Level 2: RStudio-Posit Workbench
- 4.3. Level 3: renv - for packages
- 4.4. Level 4: git - for code
- 5. More information
- 6. Hands-on practical exercise
 - 6.1. Register a free account at Posit Cloud
 - 6.2. Create a Project from git repository
 - 6.3. Choose R 3.6.x & Run Rmd
 - 6.4. Choose R 4.2.x & Run Rmd again
 - 6.5. Choose R 3.4.x & Run Rmd
 - 6.6. Additional info
- Thanks

1. Introduction - the problems (i)

TECHNOLOGY FEATURE · 24 AUGUST 2020

Challenge to scientists: does your ten-year-old code still run?

Missing documentation and obsolete environments force participants in the Ten Years Reproducibility Challenge to get creative.

Jeffrey M. Perkel

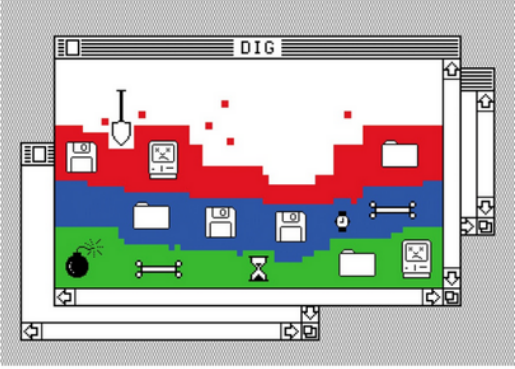



Illustration by The Pigeon Team

Perkel, J. (2020). Challenge to scientists: does your ten-year-old code still run? Nature. <https://www.nature.com/articles/d41586-020-02462-7>

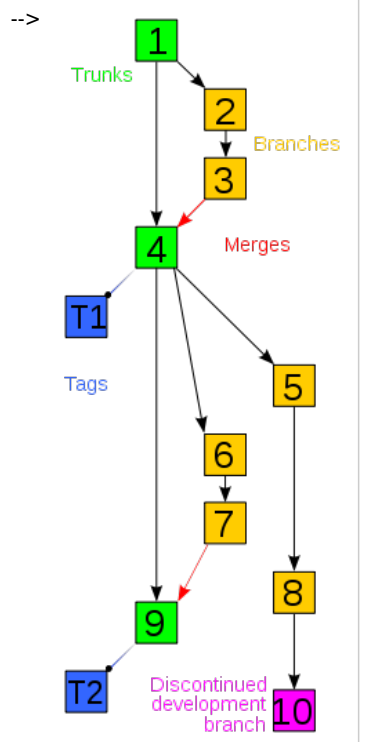
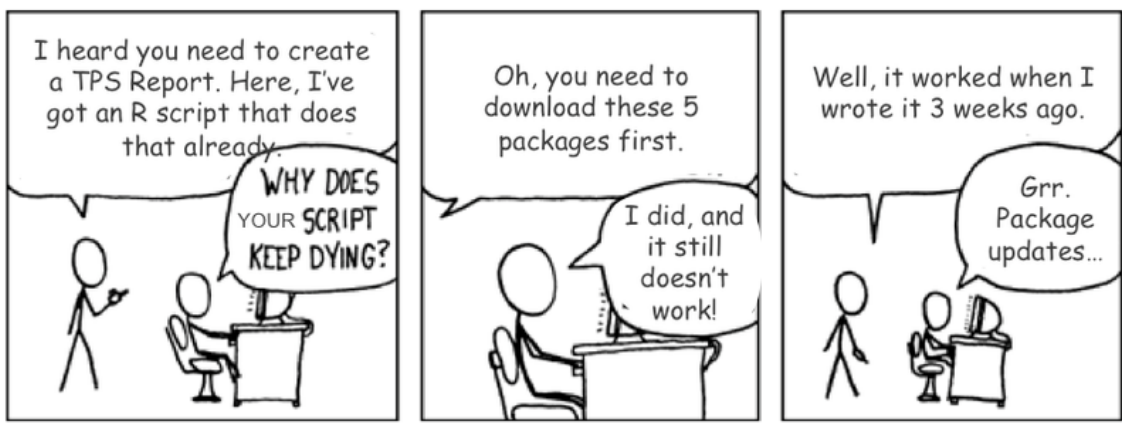
-->



From <https://www.shutterstock.com/image-illustration/3d-illustration-evolution-storage-devices-1420443290>

Obsolete Devices storing code & data --> Ease copying to new devices (legally also: copyleft, ...) + online repositories

1.1. The problems (ii)



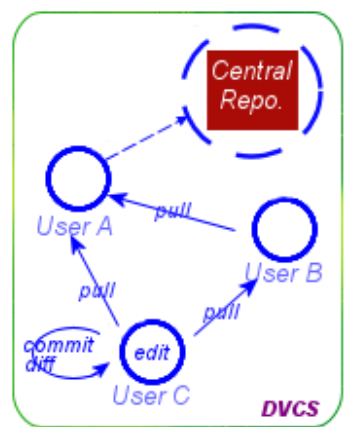
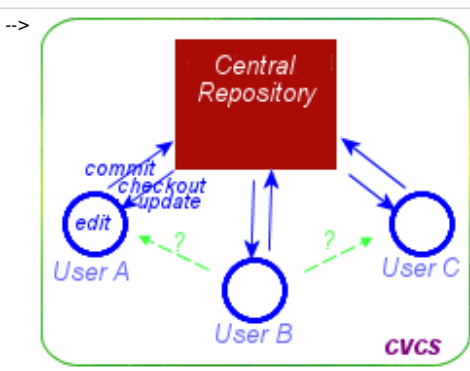
Software obsolescence and incompatible dependency versions

--> Adapt to code evolution:

- Controlling Package Versions (renv)
- VCS (git, bazaar, svn...)

VCS = Version Control Systems

1.2. The problems (iii)

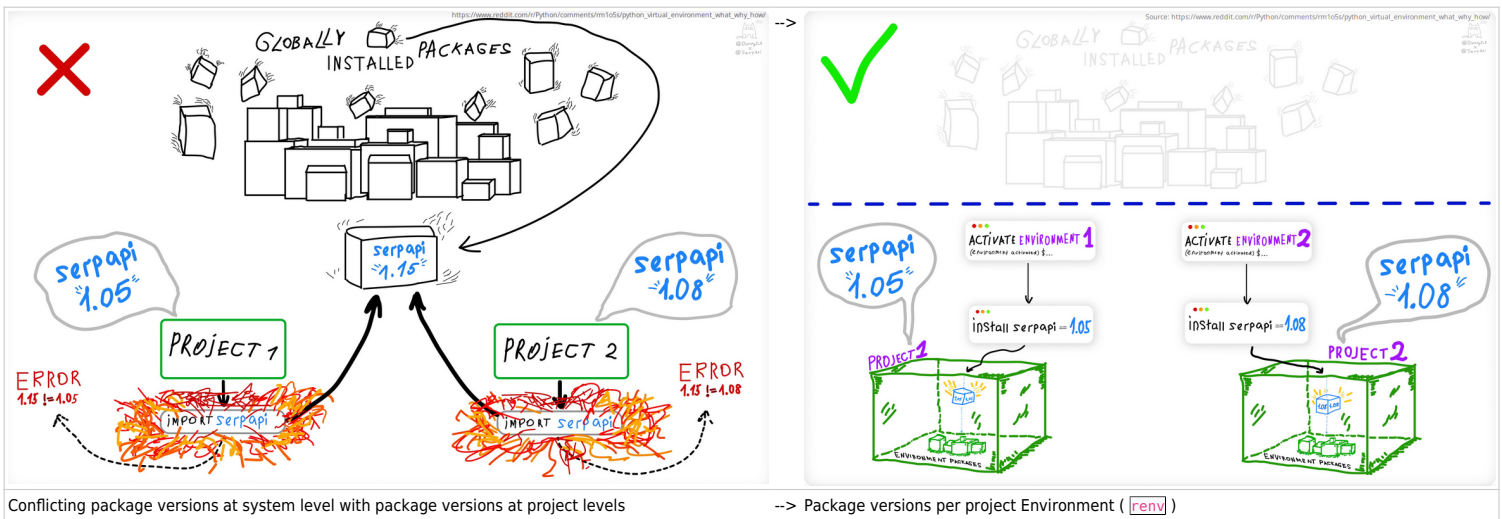


Centralization (such as **Subversion** VCS (**svn**)) may increase efficiency but it also **decreases Resilience** ("shit happens")

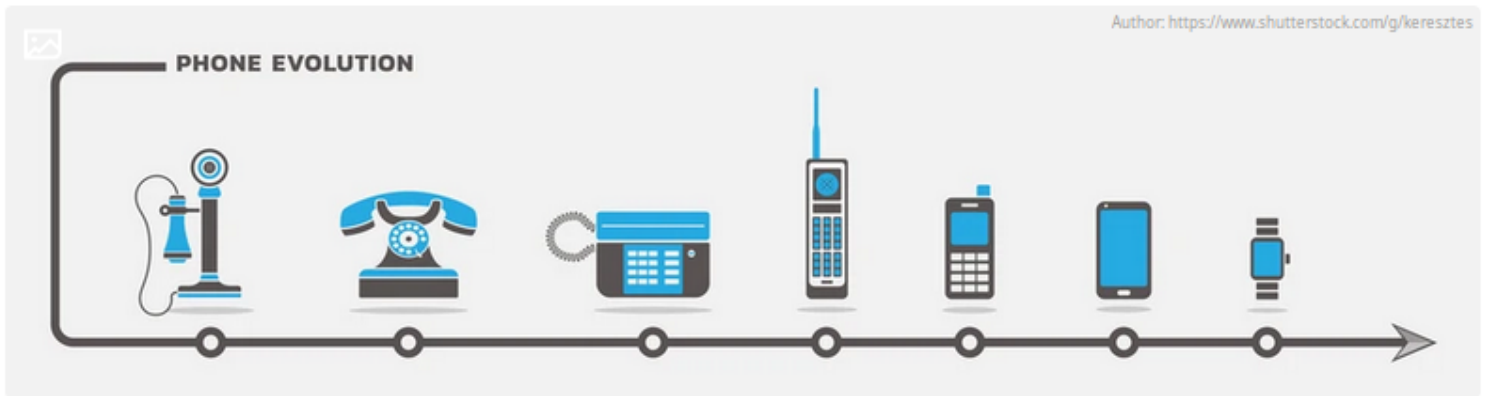
--> From **Centralized VCS** (such as **svn**) to **Decentralized VCS** (such as **git**)

VCS = Version Control Systems

1.3. The problems (iv)



1.4. The problem (v)

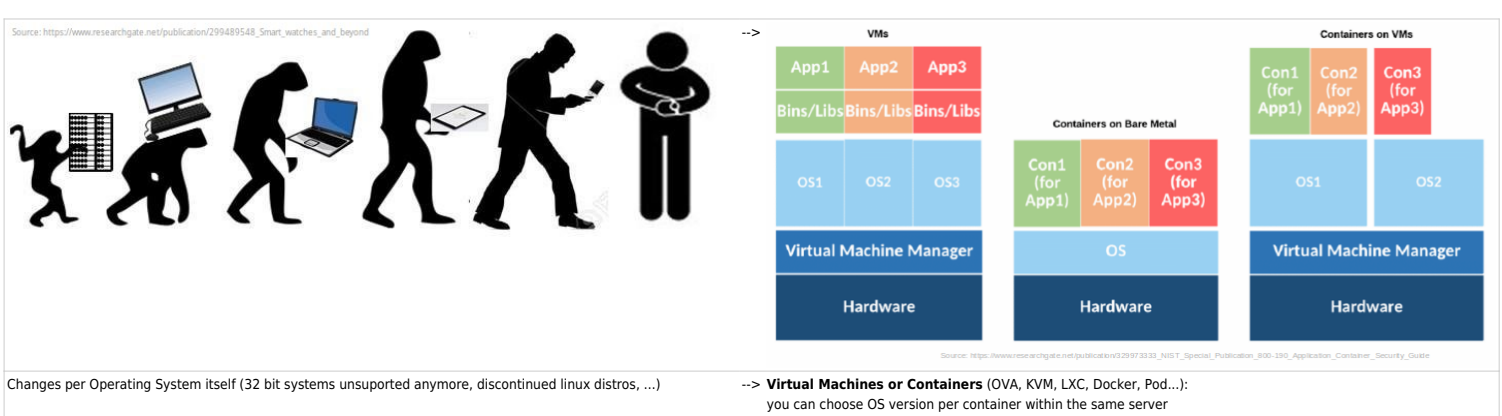


Sometimes a project was developed with a major version of a programming language (R 3.x, Python 2.x), while another project in the same server requires a different major version (R 4.x, Python 3.x)

--> **R case:** from RStudio Server to Posit Workbench (former RStudio Server Pro)
 You can choose R version per project

Python: Several approaches (conda, PyCharm, ...): see this as an example^[3].

1.5. The problem (vi)



2. Enemies of reproducibility & adaptability

Enemies of reproducibility and adaptability (in levels): Changes / Evolution / Versions!

1. **Operating system** and its **dependencies** (and their versions)
2. **Programming language** (and its version)
3. **Specific Packages** (and versions) as dependencies for your Work Project
4. **Versions** of your **own code** (algorithms and param variations, etc): lacking versioning system
5. **Readability and tidyness** of your own code / routines / scripts
6. Lack of **documentation/help resources** + steep learning curve to use it or adapt it to your context or infrastructure

3. Reproducibility & Adaptability

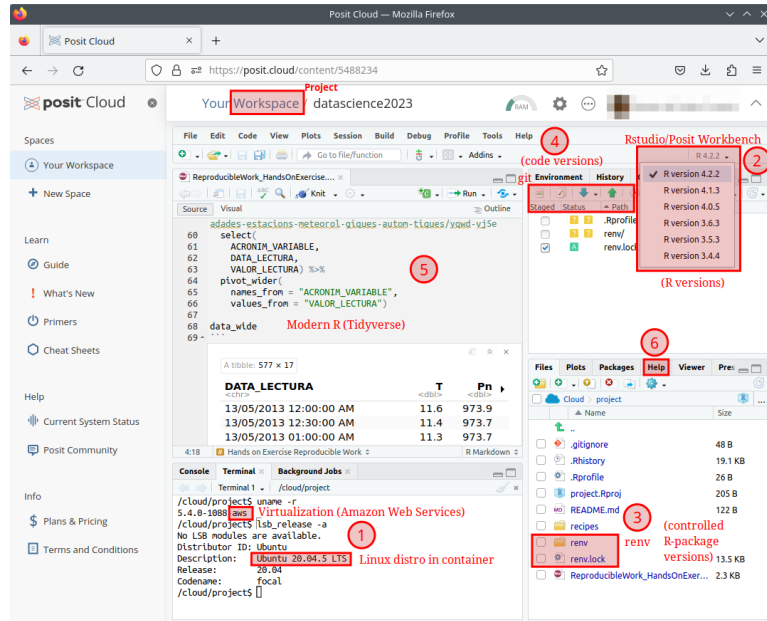
How to avoid reproducibility & adaptability enemies (in R & Python for Data Science):

<u>ISSUES</u>	<u>SOLUTIONS / WORKAROUNDS</u>
(Level 1) Versions in OS repos & critical dependencies: curl, ssl, GDAL, Java, cpp, V8...	<u>Virtual Machines</u> or <u>Containers</u> (VBox, KVM, LXC, Docker, Pod...)
(Level 2) Versions in Programming language: Python 2.x vs 3.x, R 3.x vs 4.x, ...	Python: Conda, <u>Google Colab</u> , ... R: <u>RStudio/Posit Workbench</u> General (in Linux clusters): <i>software modules</i> .
(Level 3) Versions in Specific packages	=== Py: <u>.env</u> , <u>poetry</u> R: <u>Packrat</u> , <u>Renv</u> (by versions), <u>MRAN</u> (by date)
(Level 4) Versions in Your own scripts	Decentralized VCS: <u>Git</u> (Gitlab, Github, ...), <u>Bazaar</u> (Launchpad), ... Centralized VCS: CVS, SVN (Sourceforge, ...), ... <i>VCS = Version Control system</i>
(Level 5) Tidy script content and organization	<u>Literate Coding</u> (Scripting & Coding) / Analysis - R: Rstudio Notebooks with modern R (<i>Tidyverse</i>). VS Notebooks, G-Colab, ... - Python: Jupyter Notebooks , Rstudio Notebooks, VS Notebooks, G-Colab, ... (<u>Quarto</u> Markdown and rendering for both and more)
(Level 6) Help to lower the learning curve	Documentation, Code Vignettes, Examples, Tutorials, Learning material (<u>Learnr</u>), Books (<u>bookdown</u>)...

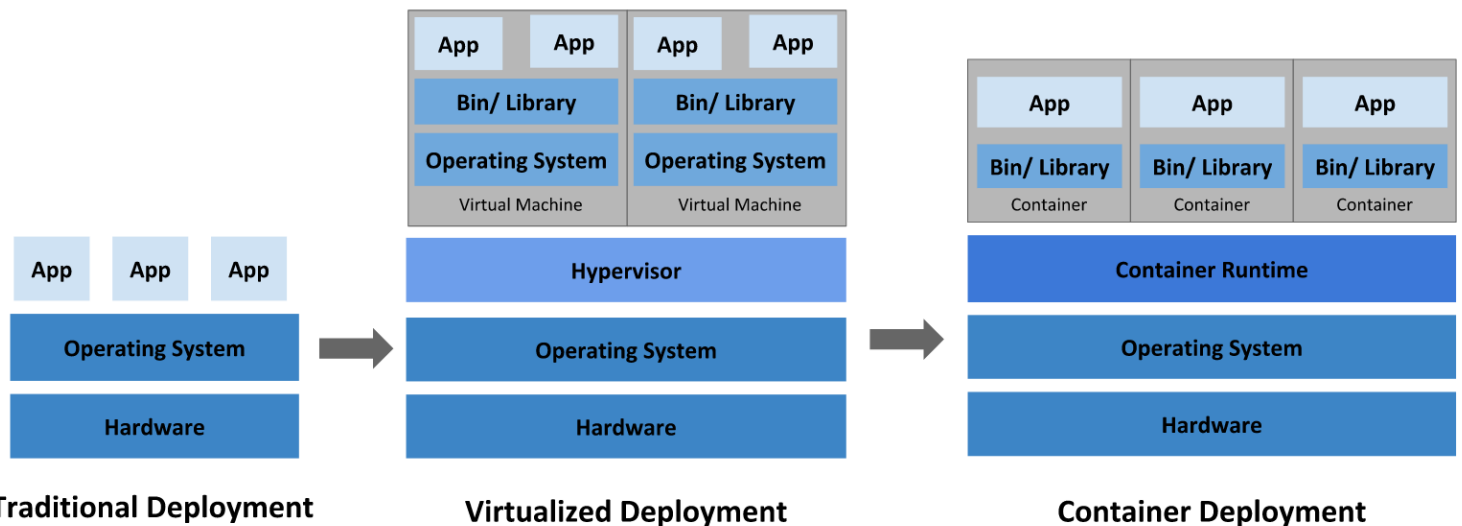
4. Reproducibility & Adaptability - Example in Posit Cloud

Example in <https://posit.cloud>^[4] (former *RStudio Server Pro*) :

- **Level 1:** A **Container** with a specific linux distro (e.g. Ubuntu Linux 20.04 Focal LTS) per project.
- **Level 2:** **RStudio/Posit Workbench** (which allows choosing R version per project)
- **Level 3:** **renv** for your R package collection (and specific versions) in your project
- **Level 4:** **git** or **svn** for your scripts in your project
- **Level 5:** YOU (*Tidyverse* is your friend)
- **Level 6:** YOU (+ helpers: [roxygen2](#), [blogdown](#), [learnr](#), [bookdown](#), ...)

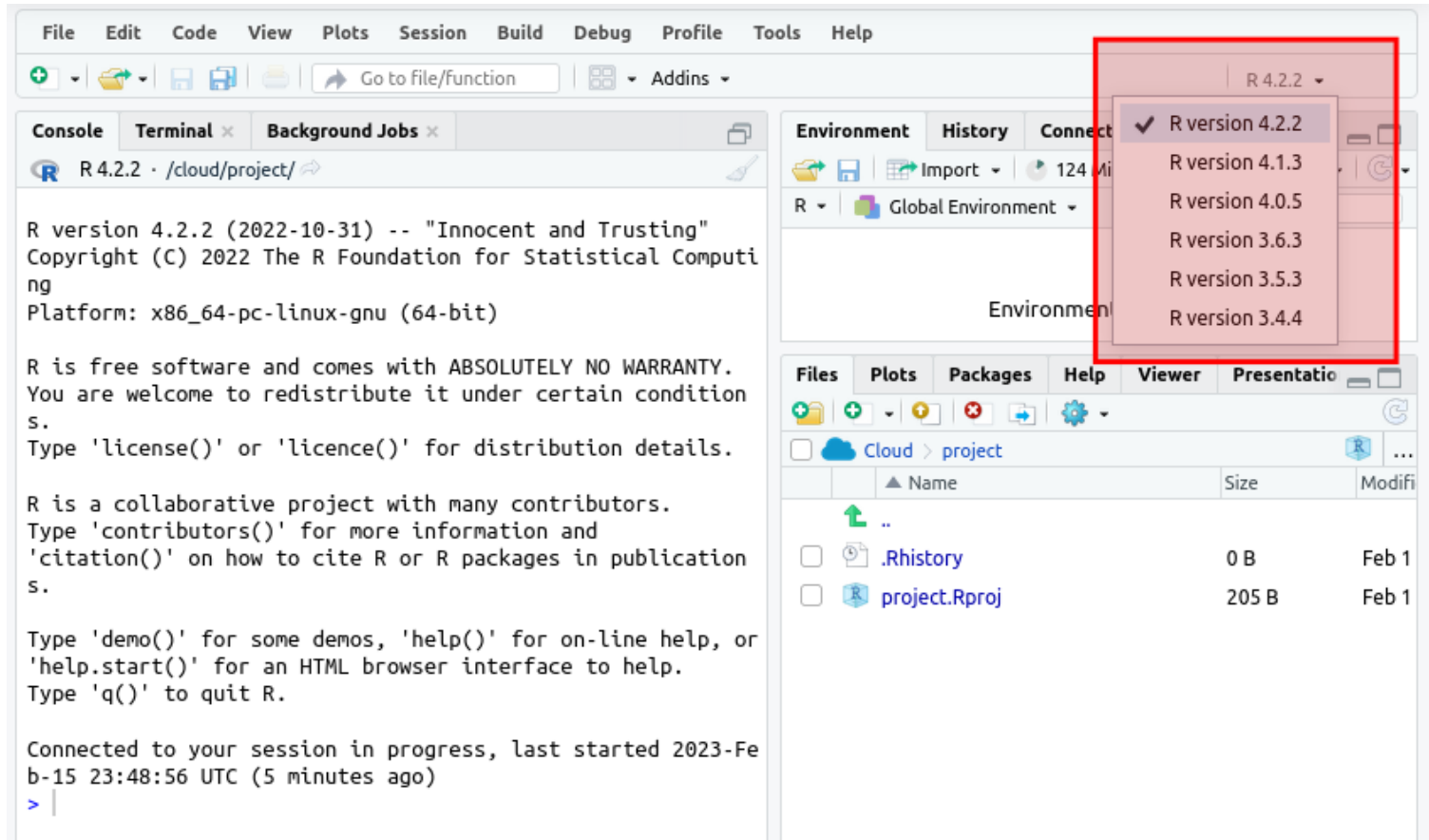


4.1. Level 1: Virtual Machines or Containers



From: <https://kubernetes.io/docs/concepts/overview/>^[5]

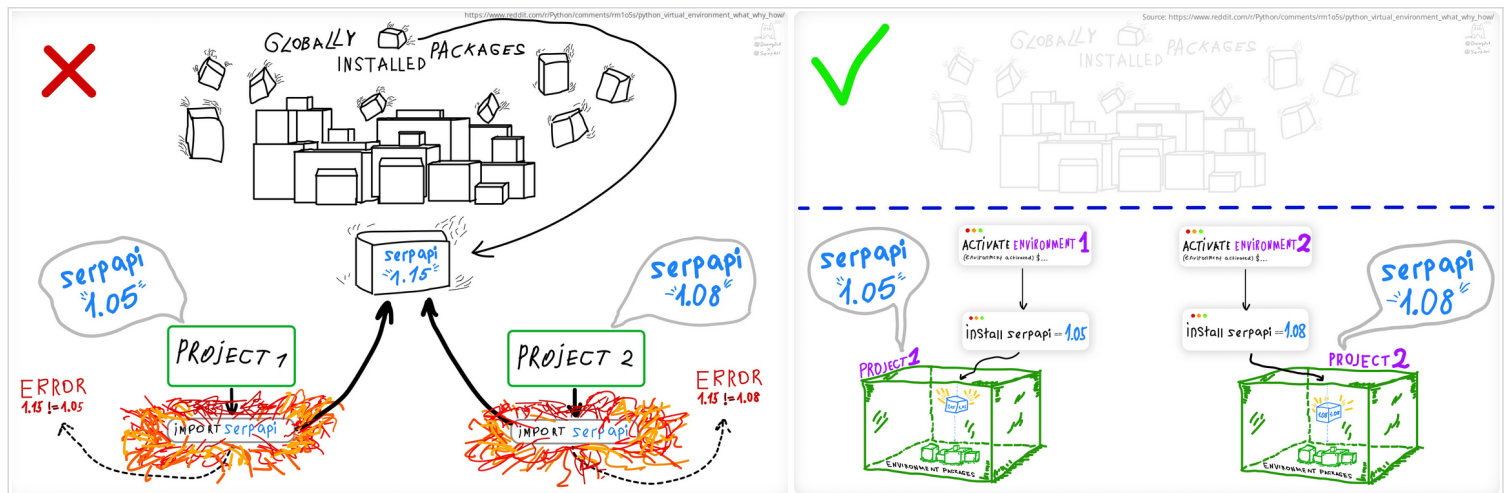
4.2. Level 2: RStudio-Posit Workbench



The screenshot shows the RStudio-Posit Workbench interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The console displays the R version 4.2.2 (2022-10-31) and the license information. The Environment pane shows the Global Environment. The Files pane shows the project directory structure, including .Rhistory and project.Rproj. A red box highlights the R version selection menu, which lists R version 4.2.2 (selected), R version 4.1.3, R version 4.0.5, R version 3.6.3, R version 3.5.3, and R version 3.4.4.

4.3. Level 3: renv - for packages

Version control in work "environments"




4.3.1. Virtual environments in R with renv



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

renv 0.16.0 Get started Reference Articles ▾ Changelog

renv



Overview

The `renv` package helps you create **reproducible environments** for your R projects. Use `renv` to make your R projects more:

- Isolated:** Installing a new or updated package for one project won't break your other projects, and vice versa. That's because `renv` gives each project its own private package library.
- Portable:** Easily transport your projects from one computer to another, even across different platforms. `renv` makes it easy to install the packages your project depends on.
- Reproducible:** `renv` records the exact package versions you depend on, and ensures those exact versions are the ones that get installed wherever you go.

Installation

Install the latest version of `renv` from CRAN with:

```
install.packages("renv")
```

Links

- [View on CRAN](#)
- [Browse source code](#)
- [Report a bug](#)

License

MIT + file [LICENSE](#)

Citation

[Citing renv](#)

Developers

Kevin Ushey
Author, maintainer
[More about authors...](#)

Dev status

lifecycle	stable
CRAN	0.16.0
R-CMD-check	failing
build	passing
codecov	unknown

4.3.2. From utils: `: sessionInfo()` to renv: `: snapshot()`
+ `renv.lock` also fails


```
utils::sessionInfo()> sessionInfo() R version 4.1.2
(2021-11-01) Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 22.04.1 LTS Matrix products:
default BLAS: /usr/lib/x86_64-linux-
gnu/blas/libblas.so.3.10.0 LAPACK: /usr/lib/x86_64-linux-
gnu/lapack/liblapack.so.3.10.0 locale: [1]
LC_CTYPE=ca_ES.UTF-8 LC_NUMERIC=C
LC_TIME=ca_ES.UTF-8 [4] LC_COLLATE=ca_ES.UTF-8
LC_MONETARY=ca_ES.UTF-8
LC_MESSAGES=ca_ES.UTF-8 [7] LC_PAPER=ca_ES.UTF-8
LC_NAME=C LC_ADDRESS=C [10] LC_TELEPHONE=C
LC_MEASUREMENT=ca_ES.UTF-8 LC_IDENTIFICATION=C
attached base packages: [1] stats graphics grDevices
datasets utils methods base other attached packages:
[1] kableExtra_1.3.4 fs_1.5.2 tictoc_1.1 lubridate_1.9.0
timechange_0.1.1 [6] janitor_2.1.0 knitr_1.40
markdown_1.3 RODBC_1.3-19 fst_0.9.8 [11]
forcats_0.5.2 stringr_1.4.1 dplyr_1. (cont.)
```

```
renv::snapshot() i ./renv.lock
```

```
{
  "R": {
    "Version": "4.1.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cloud.r-project.org"
      }
    ]
  },
  "Packages": {
    "DBI": {
      "Package": "DBI",
      "Version": "1.1.3",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash":
        "b2866e62bab9378c3cc9476a1954226b",
      "Requirements": []
    },
    "tinytex": {
      "Package": "tinytex",
      "Version": "0.42",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash":
        "7629c6c1540835d5248e6e7df265fa74",
      "Requirements": [
        "xfun"
      ]
    },
    "tzdb": {
      "Package": "tzdb",
      "Version": "0.3.0",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash":
        "b2e1cbce7c903eaf23ec05c58e59fb5e",
      "Requirements": [
        "cpp11"
      ]
    },
    "zip": {
      "Package": "zip",
      "Version": "2.2.2",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash":
        "c42bfcec3fa6a0cce17ce1f8bc684f88",
      "Requirements": []
    }
  }
}
```

```
(cont'd)0.10 purrr_0.3.5 readr_2.1.3 [16] tidyr_1.2.1
tibble_3.1.8 ggplot2_3.4.0 tidyverse_1.3.1 loaded via a
namespace (and not attached): [1] httr_1.4.4
jsonlite_1.8.3 viridisLite_0.4.1 modelr_0.1.10
assertthat_0.2.1 [6] renv_0.16.0 cellranger_1.1.0
yaml_2.3.6 pillar_1.8.1 backports_1.4.1 [11] glue_1.6.2
digest_0.6.30 rvest_1.0.3 snakecase_0.11.0
colorspace_2.0-3 [16] htmltools_0.5.3 pkgconfig_2.0.3
broom_1.0.1 haven_2.5.1 scales_1.2.1 [21]
webshot_0.5.4 svglite_2.1.0 openxlsx_4.2.5.1 rio_0.5.29
tzdb_0.3.0 [26] generics_0.1.3 ellipsis_0.3.2 withr_2.5.0
cli_3.4.1 magrittr_2.0.3 [31] crayon_1.5.2 readxl_1.4.1
evaluate_0.18 fansi_1.0.3 xml2_1.3.3 [36]
foreign_0.8-82 tools_4.1.2 data.table_1.14.4 hms_1.1.2
lifecycle_1.0.3 [41] munsell_0.5.0 reprex_2.0.2 zip_2.2.2
compiler_4.
```

(cont'd)

```
(cont'd)1.2 systemfonts_1.0.4 [46] rlang_1.0.6
grid_4.1.2 fstcore_0.9.12 rstudioapi_0.14
rmarkdown_2.18 [51] gtable_0.3.1 DBI_1.1.3 curl_4.3.3
R6_2.5.1 fastmap_1.1.0 [56] utf8_1.2.2 stringi_1.7.8
parallel_4.1.2 Rcpp_1.0.9 vctrs_0.5.0 [61] dbplyr_2.2.1
tidyselect_1.2.0 xfun_0.34 >
```

4.3.3. "Happy path"

For a reproducible environment

Commands in terminal - Computer 1

```
cd project_folder
git init
R
[obrir projecte de RStudio]
renv::init() # to initialize renv in
your code project
renv::snapshot() # to make a
snapshot "picture" of the list of R
packages used within the whole R
project and their respective package
versions
q()
git commit ...
git push
```

Commands in terminal - Computer 2

```
cd project_folder
git clone/git pull ...
R
[open same RStudio project]
renv::status() # for a report on
which steps are suggested for you to
follow
renv::restore() # to restore the
package library (with the required
package versions) for this project
[continue working in/developing your
code]
renv::snapshot() # to make a new
snapshot "picture" (in case there
are new packages and/or versions or
R packages newer or older in use in
your project ;- )
q()
git commit ...
```

```
git push
```

4.3.4. Infraestructure

Projects with `renv` write and use these files in order to work:

File	Use
<code>.Rprofile</code>	Used to activate <code>renv</code> for new R sessions launched in the project.
<code>renv.lock</code>	The lockfile, describing the state of your project's library at some point in time.
<code>renv/activate.R</code>	The activation script run by the project .Rprofile .
<code>renv/library</code>	The private project library.
<code>renv/settings.dcf</code>	Project settings - see <code>?settings</code> for more details.

By default, `renv` uses a package memory-cache here:

Platform	Location
Linux	<code>~/.local/share/renv</code>
macOS	<code>~/Library/Application Support/renv</code>
Windows	<code>%LOCALAPPDATA%/renv</code>

4.3.5. Advanced use

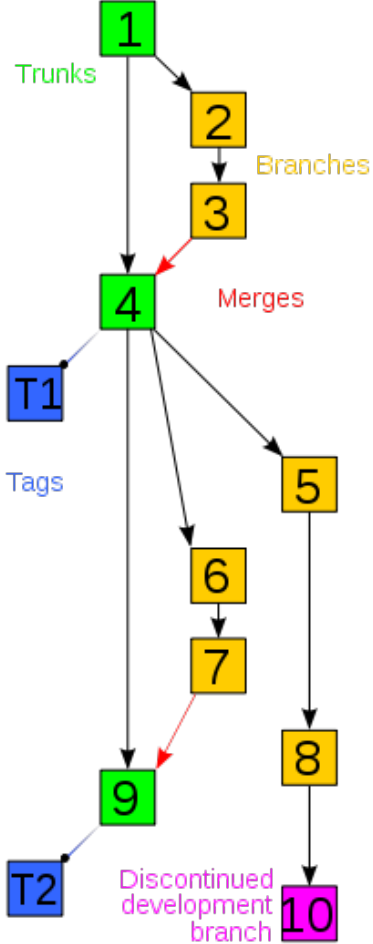


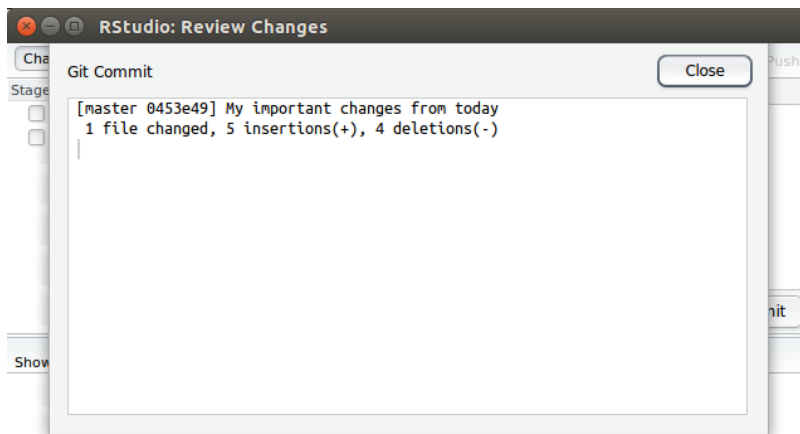
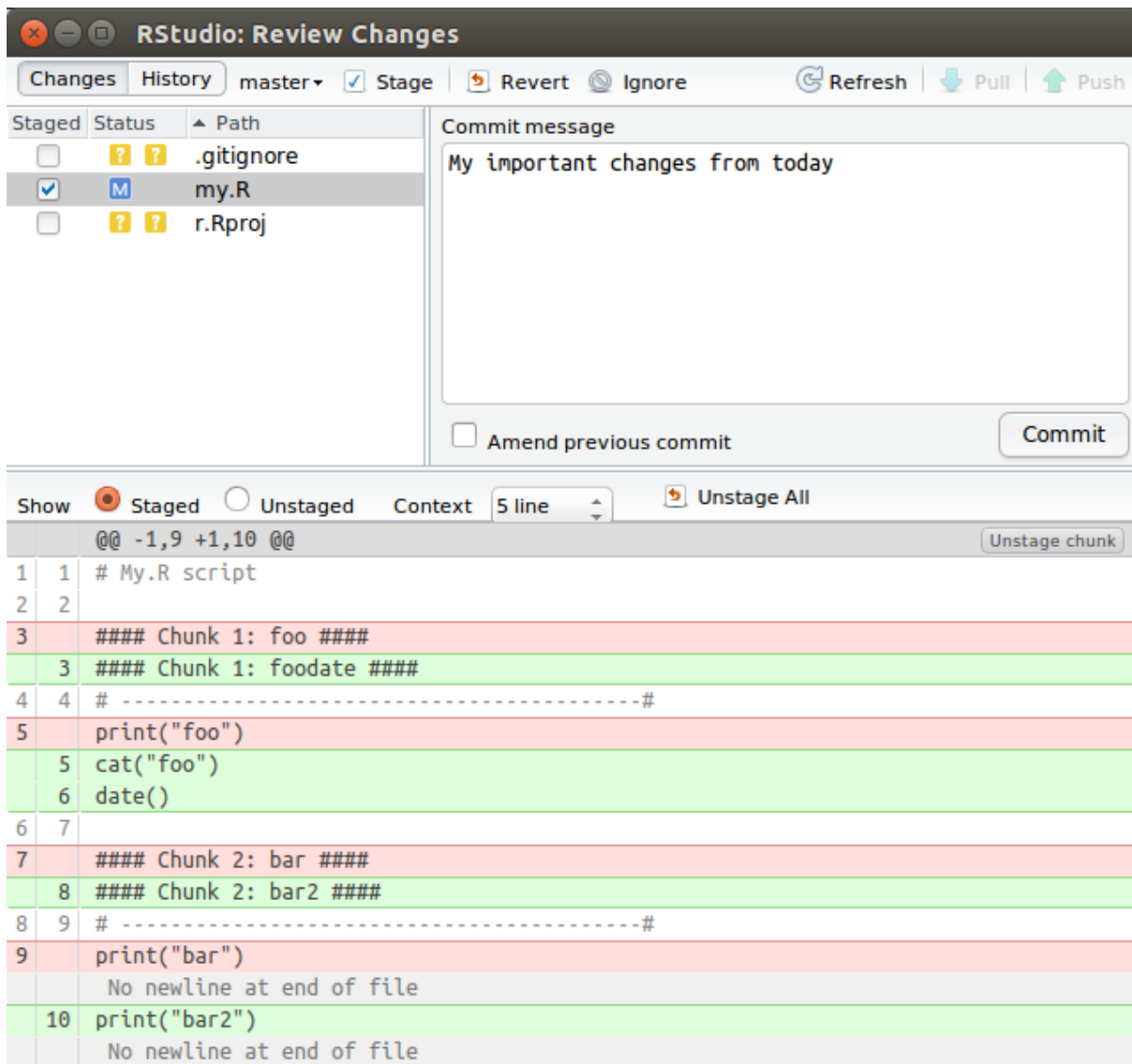
```
renv::install("packagename", version="0.1") # to install old versions from a
package (useful also for discontinued packages in CRAN!). See possible package-
version numbers at https://cran.r-project.org/src/contrib/Archive/yourpackage/
renv::record("packagename", version="0.1") # to save at renv.lock the specific
version you need for this package
renv::deactivate() # to temporarily deactivate renv in your project
renv::activate() # to reactivate renv in your project
renv::equip() # for special installations in MS Windows
vignette("docker", package = "renv") # for a combined use with Docker
vignette("collaborating", package = "renv") # to improve collaborative use in work
teams
```

And much more. See:

- <https://rstudio.github.io/renv/articles/renv.html>^[6]
- <https://solutions.posit.co/envs-pkgs/environments/>^[7]

4.4. Level 4: git - for code





See: <https://gitlab.com/radup/curs-r-introduccio/>^[8] > Folder "codi"^[9] > **10.compartir.via.git.Rmd** (or .pdf^[10])

See also my own git recipes over some years, github cheatsheet, ...: <https://seeds4c.org/git>^[11]

5. More information

Work Environments in R

- <https://solutions.posit.co/envs-pkgs/environments/>^[12]

Videos

- An Introduction to Reproducible Research Practices. 29 d'abr. 2022. John Little. Duke University. Video^[13]
- Designing a Reproducible Workflow with R and GitHub. John Little. 22 de nov. 2021 Video^[14] | Tutorial^[15]
- The workflowr R package: a framework for reproducible and collaborative data science. 13 de jul. 2018. R Consortium. Video^[16]
- Kevin Ushey | renv: Project Environments for R | RStudio (2020). Posit PBC.. 20 de des. 2020. Video^[17]

R Packages

[renv](#)^[18] | [workflowr](#)^[19] | [learnr](#)^[20] | [roxygen2](#)^[21] | [Tidyverse](#)^[22]

Free Work environments for Collaborative Data Science with R & Python

- <https://posit.cloud/plans/free>^[23]

Additional tutorial with big data to follow on site (R Cloud)

- Danielle Navarro. 2022. "Using Amazon S3 with R"^[24] March 17, 2022.

Papers

- Wallach JD, Boyack KW, Ioannidis JPA. (2018) Reproducible research practices, transparency, and open access data in the biomedical literature, 2015–2017. PLoS Biol 16 (11): e2006930. <https://doi.org/10.1371/journal.pbio.2006930>^[25]
- Leek JT, Peng RD. Opinion: Reproducible research can still be wrong: adopting a prevention approach. Proc Natl Acad Sci U S A. 2015 Feb 10;112(6):1645-6. doi: 10.1073/pnas.1421412111. PMID: 25670866; PMCID: PMC4330755

6. Hands-on practical exercise

The screenshot displays the Posit Cloud workspace for 'datascience2023'. The main editor shows an R script with the following code:

```

60 select(
61   ACRONIM_VARIABLE,
62   DATA_LECTURA,
63   VALOR_LECTURA) %>%
64   pivot_wider(
65     names_from = "ACRONIM_VARIABLE",
66     values_from = "VALOR_LECTURA")
67
68 data_wide
69

```

Below the script, a tibble is displayed with 577 rows and 17 columns. The visible columns are DATA_LECTURA, T, and Pn.

DATA_LECTURA	T	Pn
13/05/2013 12:00:00 AM	11.6	973.9
13/05/2013 12:30:00 AM	11.4	973.7
13/05/2013 01:00:00 AM	11.3	973.7

The console shows the output of the following commands:

```

/ccloud/project$ uname -r
5.4.0-1088-aws
/ccloud/project$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.5 LTS
Release:        20.04
Codename:       focal
/ccloud/project$

```

The file explorer on the right shows the project structure:

- Cloud > project
 - ..
 - .gitignore (48 B)
 - .Rhistory (19.1 KB)
 - .Rprofile (26 B)
 - project.Rproj (205 B)
 - README.md (122 B)
 - recipes
 - renv
 - renv.lock (13.5 KB)
 - ReproducibleWork_HandsOnExer... (2.3 KB)

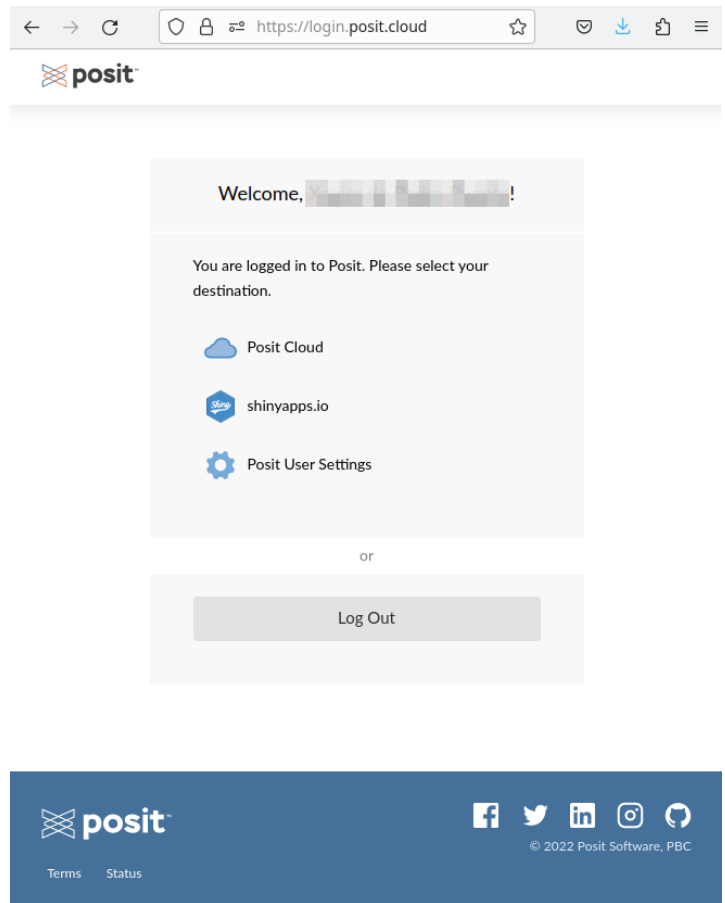
6.1. Register a free account at Posit Cloud

You can do so at:

- <https://posit.cloud/plans/free>^[26]

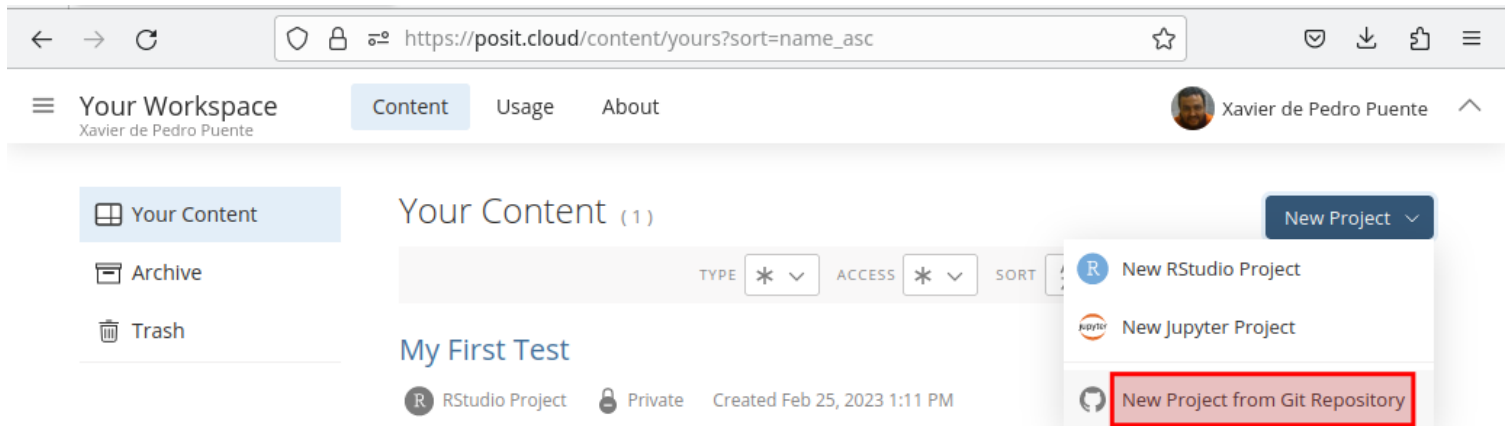
You will need to click on a link sent to your email inbox to validate your account.

Once done, you'll see something like:



6.2. Create a Project from git repository

Enter Posit cloud and click at **New Project > New Project from Git Repository**



6.2.1. Visit gitlab to get clone url

Visit this code project in gitlab to get the project clone url:

<https://gitlab.com/xavidp/datascience2023>^[27]

The screenshot shows the GitLab interface for a repository named 'DataScience2023'. The URL in the browser is `https://gitlab.com/xavidp/datascience2023`. The repository has 5 commits, 1 branch, 0 tags, and 236 KB of project storage. A recent commit titled 'Base Rmd file' is shown, authored by Xavier de Pedro 9 minutes ago. The 'Clone' button is highlighted with a red box. A dropdown menu is open, showing options to clone with SSH, HTTPS, or open in an IDE. The 'Clone with HTTPS' option is highlighted with a red box, and the 'Open in your IDE' section is also visible.

Name	Last commit
<code>.gitignore</code>	Afegit rproj
<code>README.md</code>	Update 2 files
<code>ReproducibleWork_HandsOn...</code>	Base Rmd file

6.2.2. Create project from git repo

Paste it in the Posit cloud popup window and click at OK:

The screenshot shows the Posit Cloud interface. The browser URL is `https://posit.cloud/content/yours?sort=name_asc`. The user is logged in as 'Xavier de Pedro Puente'. A dialog box titled 'New Project from Git Repository' is open, with the URL `https://gitlab.com/xavidp/datascience2023.git` entered in the 'URL of your Git Repository' field. The 'OK' button is highlighted.

6.3. Choose R 3.6.x & Run Rmd

The screenshot shows the Posit Cloud interface in a Mozilla Firefox browser. The URL is <https://posit.cloud/content/5488234>. The workspace is named "datascience2023". The R version is set to "R 3.6.3" (indicated by a red circle 1). The "Run" menu is open, showing options like "Run Selected Line(s)", "Run Current Chunk", "Run Next Chunk", "Run Setup Chunk", "Run Setup Chunk Automatically", "Run All Chunks Above", "Run All Chunks Below", "Restart R and Run All Chunks", "Restart R and Clear Output", and "Run All" (indicated by a red circle 4). The "Run All" button is highlighted with a red box. The console shows the R prompt and the message "Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R." The file explorer shows the project files, including "ReproducibleWork_HandsOnExercise.Rmd" (indicated by a red circle 2).

6.3.1. Install dependencies also

The screenshot shows the Posit Cloud interface with a warning message: "Packages markdown and knitr required but are not installed. Install Don't Show Again". The "Run" menu is open, and the "Run All" button is highlighted with a red box. The console shows the R prompt and the message "Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R." The file explorer shows the project files, including "ReproducibleWork_HandsOnExercise.Rmd".

```

11
12 # Session Reproducible Work
13
14 Monday Feb 27, 2023. IL3-UB.
15
16 Related to:
17 https://seeds4c.org/reproduciblework2023
18
4:5 # Hands on Exercise Reproducible Work R Markdown

```

```

Install R packages 0:05
* DONE (base64enc)
* installing *binary* package 'mime' ...
* DONE (mime)
* installing *binary* package 'ellipsis' ...
* DONE (ellipsis)
* installing *binary* package 'cachem' ...
* DONE (cachem)

```

6.3.2. Running Rmd will perform GNU/Linux system commands also

GNU/Linux system commands will usually be much more efficient in memory & cpu

It helps to prevent RAM bottlenecks with just 1Gb RAM on Posit Cloud Free plan

(while csv file from reduced meteorological dataset is already 0.5 Gb).

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for installing the 'readr' package and downloading a dataset. Lines 27-29 are highlighted in red:


```

27 system("wget http://cloud.seeds4c.org/data_smc.csv.bz2")
28 system("bunzip2 data_smc.csv.bz2 -k")
29 system("cat data_smc.csv | head -n1000001 > data_subset.csv")

```
- Console:** Shows the execution of the code, including the message "Loading required package: readr" and the output of the 'read_csv' function:


```

> data <- read_csv("data_subset.csv")
Rows: 1000000 Columns: 8 — Column specification

```
- Files Panel:** Shows a file explorer view of the project directory. Files are listed with their sizes:

Name	Size
..	48 B
.gitignore	0 B
.Rhistory	0 B
data_smc.csv.bz2	50.2 MB
project.Rproj	205 B
README.md	122 B
ReproducibleWork_HandsOnExer...	629 B
data_smc.csv	613.3 MB
data_subset.csv	61.3 MB

6.3.3. Display raw data

Variables are in numeric codes (not easily readable by humans in a semantic way). We lack some variable names (or acronyms at least) for readability.

ID	CODI_ESTACIO	CODI_VARIABLE	DATA_LECTURA	DATA_EXTREM
1	XK721205132330	XK	72 12/05/2013 11:30:00 PM	12/05/2013 11:30:00 PM
2	XK361205132330	XK	36 12/05/2013 11:30:00 PM	NA
3	XK381205132330	XK	38 12/05/2013 11:30:00 PM	NA
4	XK321205132330	XK	32 12/05/2013 11:30:00 PM	NA
5	XK401205132330	XK	40 12/05/2013 11:30:00 PM	12/05/2013 11:30:00 PM
6	XK421205132330	XK	42 12/05/2013 11:30:00 PM	12/05/2013 11:51:00 PM
7	XK331205132330	XK	33 12/05/2013 11:30:00 PM	NA
8	XK441205132330	XK	44 12/05/2013 11:30:00 PM	12/05/2013 11:30:00 PM
9	XK031205132330	XK	3 12/05/2013 11:30:00 PM	12/05/2013 11:51:00 PM
10	XK301205132330	XK	30 12/05/2013 11:30:00 PM	NA
11	XK311205132330	XK	31 12/05/2013 11:30:00 PM	NA
12	XL031205132330	XL	3 12/05/2013 11:30:00 PM	12/05/2013 11:51:00 PM
13	XL301205132330	XL	30 12/05/2013 11:30:00 PM	NA

6.3.4. Transform in tidy way (i)

```

34
35 `r`
36 # Get the description of the variable codes
37 # From here: https://analisi.transparenciacatalunya.cat/Medi-Ambient/Metadades-variables-meteorol-giques/4fb2-n3yi/data
38 variables <- read_csv("https://analisi.transparenciacatalunya.cat/api/views/4fb2-n3yi/rows.csv?accessType=DOWNLOAD&sorting=true")
39 `r`

Rows: 26 Columns: 6 — Column specification —————
Delimiter: ","
chr (4): NOM_VARIABLE, UNITAT, ACRONIM, CODI_TIPUS_VAR
dbl (2): CODI_VARIABLE, DECIMALS
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.

40
41 `r`
42 # We prepare a small dataframe from the variable definition to join on the smc data frame
43 variables.to.join <- variables %>%
44   select(CODI_VARIABLE, ACRONIM) %>%
45   arrange(CODI_VARIABLE)
46
47 variables.to.join
48 `r`

A tibble: 26 x 2
  CODI_VARIABLE ACRONIM
  <dbl> <chr>
1 1 Px
2 2 Pn
3 3 HRx
...
30 VV10

```

6.3.5. Transform in tidy way (ii) - result

```

49
50 ~ ``{r}
51 # Let's join variable df on to the data df
52 data <- left_join(data, variables.to.join) %>%
53   rename(ACRONIM_VARIABLE = ACRONIM)
54 ~ ``

```

Joining, by = "CODI_VARIABLE"

```

55
56 ~ ``{r}
57 # Let's convert the source data frame (which is long shape, as database) into a wide shape (table like, with meteorological variables as
58 columns) while selecting just one meteorological station as an example
59 data_wide <- data %>%
60   filter(CODI_ESTACIO == "D5") %>% # D5 corresponds to "Barcelona Observatori Fabra" Meteorological Observatory (at Collserola Mountain)
61   https://analisi.transparenciacatalunya.cat/Medi-Ambient/Metadades-estacions-meteorol-giques-auton-tiques/vqwd-vj5e
62   select(
63     ACRONIM_VARIABLE,
64     DATA_LECTURA,
65     VALOR_LECTURA) %>%
66   pivot_wider(
67     names_from = "ACRONIM_VARIABLE",
68     values_from = "VALOR_LECTURA")
69 ~ ``

```

A tibble: 577 x 17

DATA_LECTURA <chr>	T <dbl>	Pn <dbl>	Tn <dbl>	HR <dbl>	HRn <dbl>	HRx <dbl>	VV10 <dbl>	DV10 <dbl>	VVx10 <dbl>
13/05/2013 12:00:00 AM	11.6	973.9	11.4	91	91	92	2.0	238	2.7
13/05/2013 12:30:00 AM	11.4	973.7	11.4	90	90	91	1.5	238	2.4
13/05/2013 01:00:00 AM	11.3	973.7	11.3	89	87	91	1.1	174	2.3
13/05/2013 01:30:00 AM	11.3	973.6	11.3	89	88	91	1.5	209	2.4

6.3.6. Last code chunks

```

70
71 ~ ``{r}
72 # Save resulting dataset to disk
73 write_csv(data_wide, "data_subset_d5_wide.csv")
74 ~ ``
75
76
77 ~ ``{r}
78 # Produce a simple R version of this R Markdown document
79 knitr::purl("ReproducibleWork_HandsOnExercise.Rmd", documentation=2)
80 ~ ``

```

[1] "ReproducibleWork_HandsOnExercise.R"

```

81
82

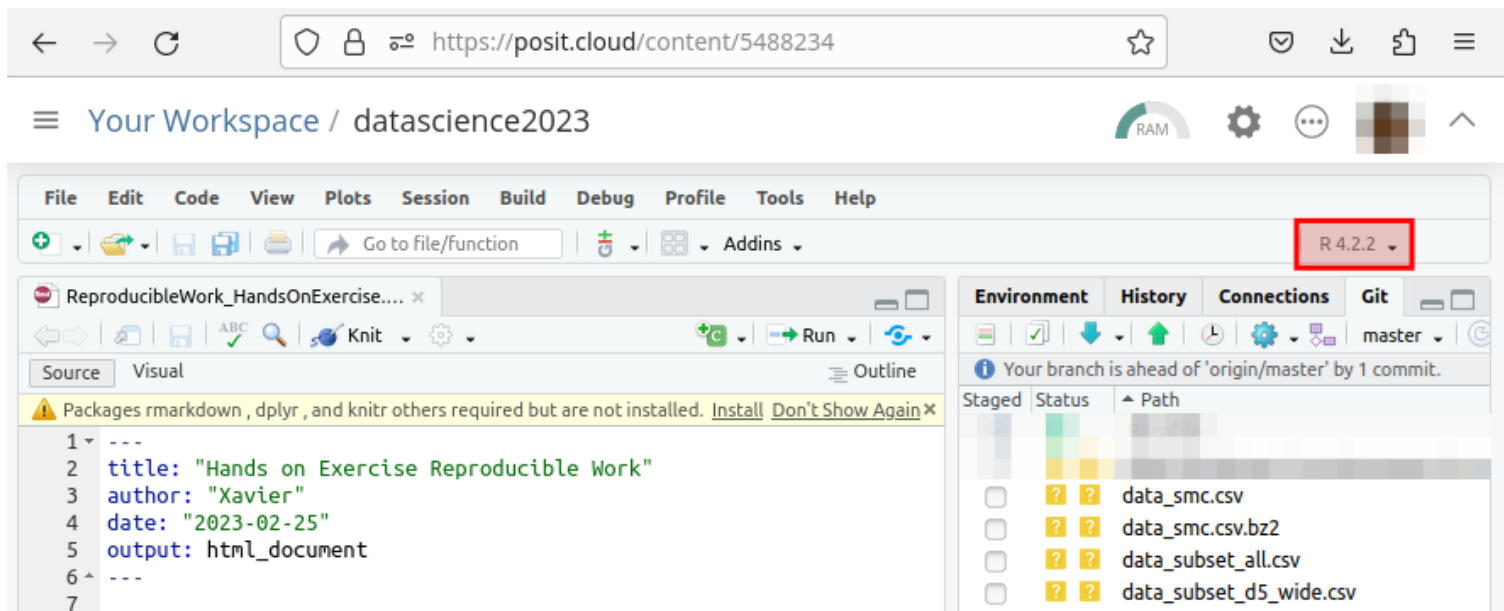
```

4:18 # Hands on Exercise Reproducible Work ↕

6.4. Choose R 4.2.x & Run Rmd again

Repeat the previous steps but in a R 4.2.x environment: install dependent R packages again... (new environment, but still installing from CRAN repos). renv not needed in this case still (lucky you!).

So far, so good.



6.5. Choose R 3.4.x & Run Rmd

Now let's touch some issues with R package versions in a R 3.4.x environment

Running Rmd will fail at some package installations

- `dplyr` installation fails
- `readr` is reported as unavailable in R 3.4.4
- `tidyr` installation also fails (as well as `purrr`)

Solution

In this case, the solution involves finding some valid previous package version for each conflicting R package, and using this type of commands:

- `renv::init()`
- `renv::install("packagename@x.y.z")` # being x.y.z a valid package version number, as taken from <https://cran.r-project.org/src/contrib/Archive/packagename/>^[28]
- `renv::record("packagename@x.y.z")`
- `renv::snapshot()` # after all packages installed without any more issues

```

Console Terminal Background Jobs
R 3.4.4 . /cloud/project/
> renv::init()
Error in loadNamespace(name) : there is no package called 'renv'
> install.packages("renv")
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/3.4.4'
(as 'lib' is unspecified)
trying URL 'http://rspm/default/_linux_/focal/latest/src/contrib/renv_0.16.0.tar.gz'
Content type 'application/x-gzip' length 1878804 bytes (1.8 MB)
=====
downloaded 1.8 MB

* installing *binary* package 'renv' ...
* DONE (renv)

The downloaded source packages are in
'/tmp/RtmpzvsNWy/downloaded_packages'
> renv::init()
* Initializing project ...
* Discovering package dependencies ... Done!
* Copying packages into the cache ... Done!
The following package(s) will be updated in the lockfile:

# RSPM =====
- R6 [ * -> 2.5.1 ]
- base64enc [ * -> 0.1-3 ]

```

6.5.1. Use `renv.lock` recipe (i)

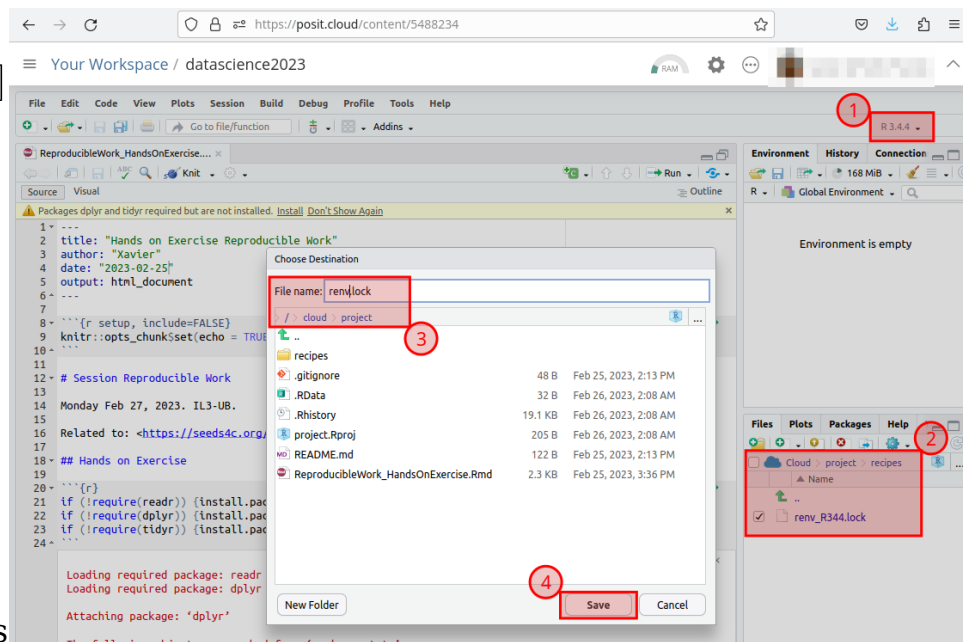
Let's get `renv` to the rescue.

Once somebody solved these issues, and found a valid recipe of package versions for this environment, a file `./renv.lock` will have been produced in the project root

folder after running the command `renv::snapshot()`

I did this already, and I uploaded the produced `renv.lock` file to the manually created `./recipes/` folder in this project as a backup for you (as `renv_R344.lock`).

You can then copy now the `./recipes/renv_R344.lock` file provided in the project as `./renv.lock` in the project root folder, for `renv` to be able use it.

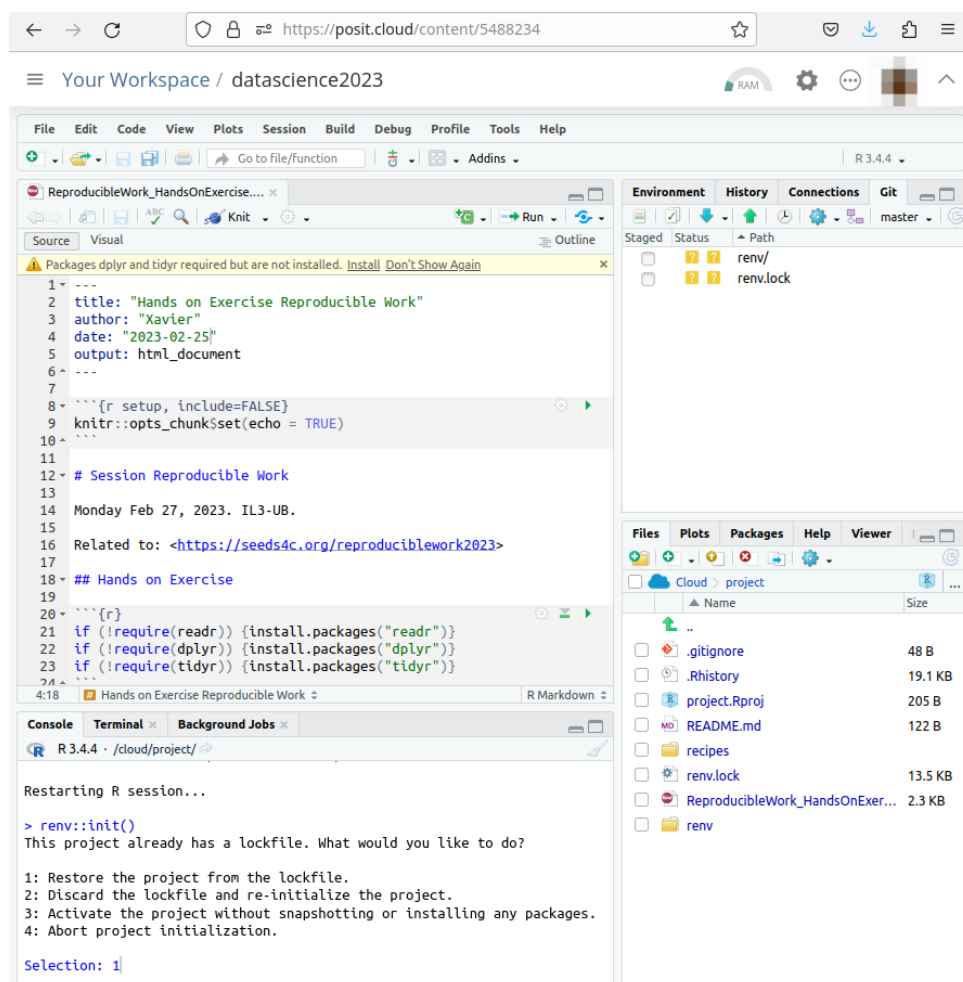


6.5.2. Use renv.lock recipe (ii)

Run `renv::init()` in the R console.

Choose restore the renv.lock package versions:

"1. Restore the project from the lockfile"

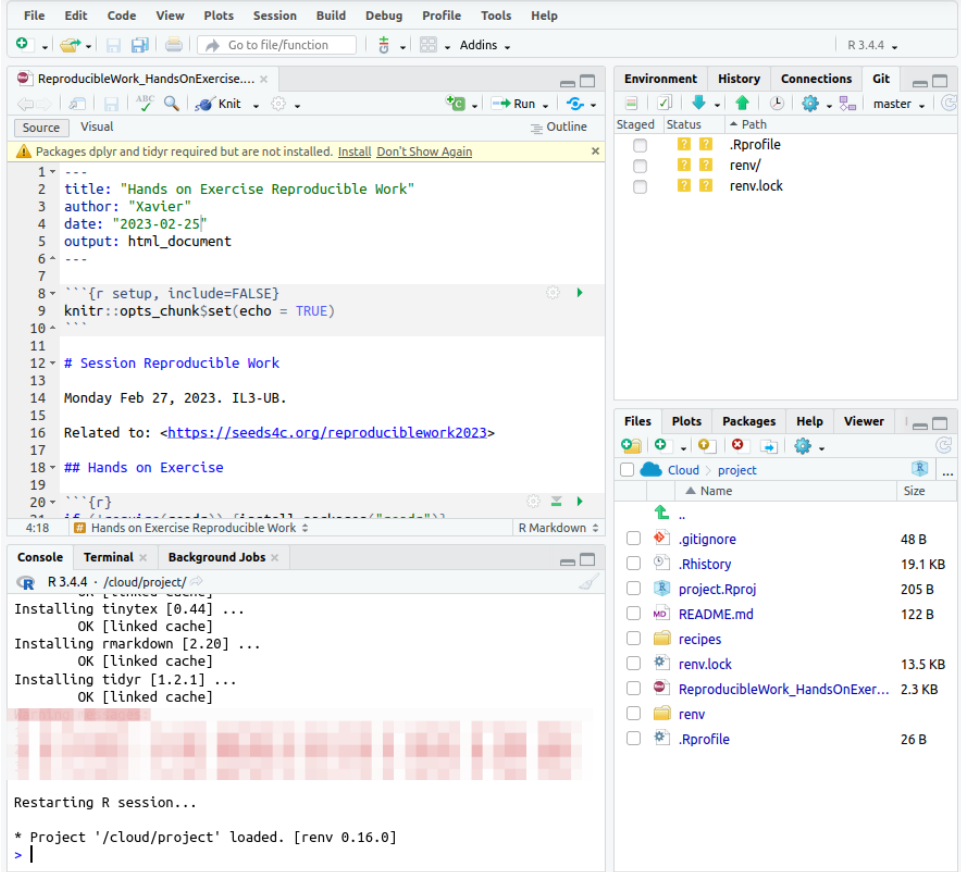


6.5.3. Use renv.lock recipe (iii)

You will be ready to go with minimum human intervention.

All R packages will be installed in the background to their required package versions, following the recipe that someone created for R 3.4.4. already.

The key file is the **renv.lock** file.



The screenshot shows the RStudio interface with the following components:

- Source Editor:** Displays R Markdown code for a project titled "Hands on Exercise Reproducible Work". The code includes a title, author ("Xavier"), date ("2023-02-25"), output type ("html_document"), and a session recipe for R 3.4.4. The recipe specifies the installation of packages `dplyr` and `tidyr`. The code also includes a session title and a link to the project's GitHub repository.
- Environment Panel:** Shows the project's environment with the following packages listed:

Staged	Status	Path
<input type="checkbox"/>	?	.Rprofile
<input type="checkbox"/>	?	renv/
<input type="checkbox"/>	?	renv.lock
- Files Panel:** Shows the project's file structure with the following files and sizes:

Name	Size
..	
.gitignore	48 B
.Rhistory	19.1 KB
project.Rproj	205 B
README.md	122 B
recipes	
renv.lock	13.5 KB
ReproducibleWork_HandsOnExer...	2.3 KB
renv	
.Rprofile	26 B
- Console:** Shows the output of the R session, including the installation of `tinytex`, `rmarkdown`, and `tidyr`. The output indicates that the packages were installed successfully from the linked cache. The console also shows the message "Restarting R session..." and the message "* Project '/cloud/project' loaded. [renv 0.16.0]".

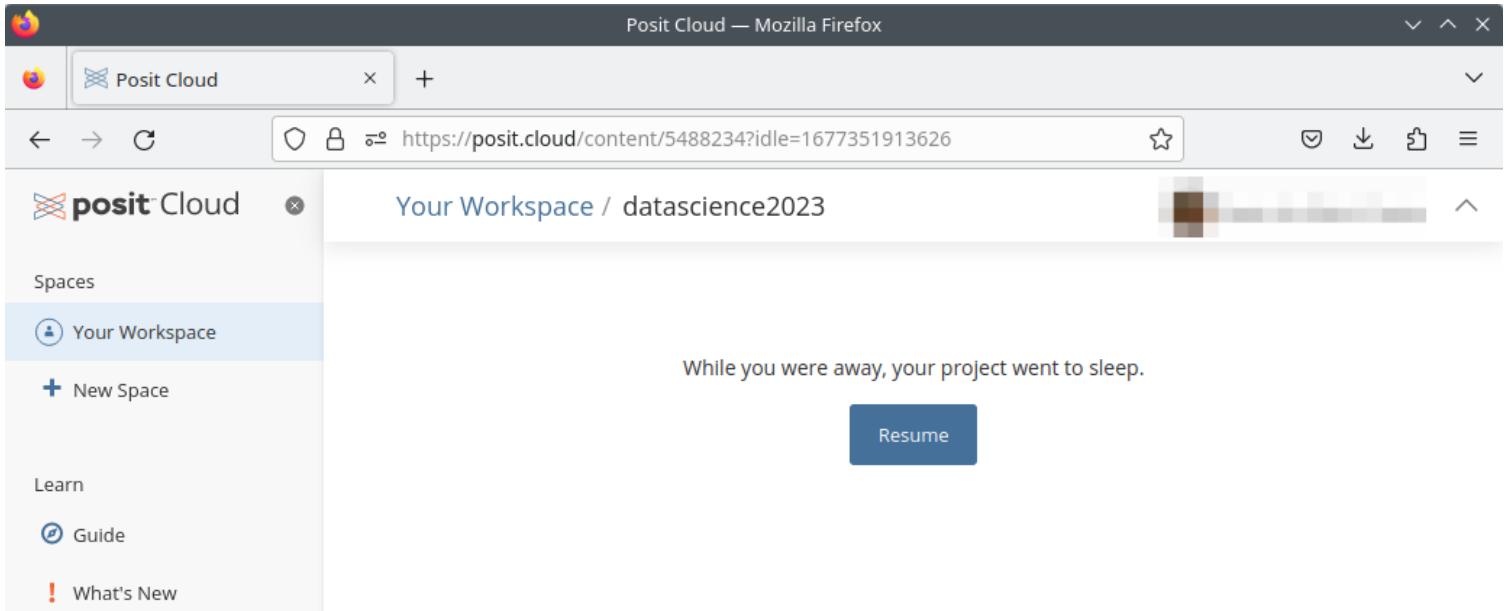
6.5.4. Use renv.lock recipe (iv) - finished

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Shows R Markdown code for a document titled "Hands on Exercise Reproducible Work" by "Xavier" dated "2023-02-25". The code includes a Knit setup, session information, and R code to read data and generate a report.
- Console:** Shows the execution output, including the message "processing file: ReproducibleWork_HandsOnExercise.Rmd" and "output file: ReproducibleWork_HandsOnExercise.R". The final output is "[1] \"ReproducibleWork_HandsOnExercise.R\"".
- Environment/History:** Lists files generated by the project, such as .Rprofile, ReproducibleWork_HandsOnExercise.R, and various data files.
- Files Panel:** Shows the project's file structure, including .gitignore, .Rhistory, project.Rproj, README.md, recipes, renv.lock, and the R Markdown file.

6.6. Additional info

Project (Container) goes to sleep on inactivity

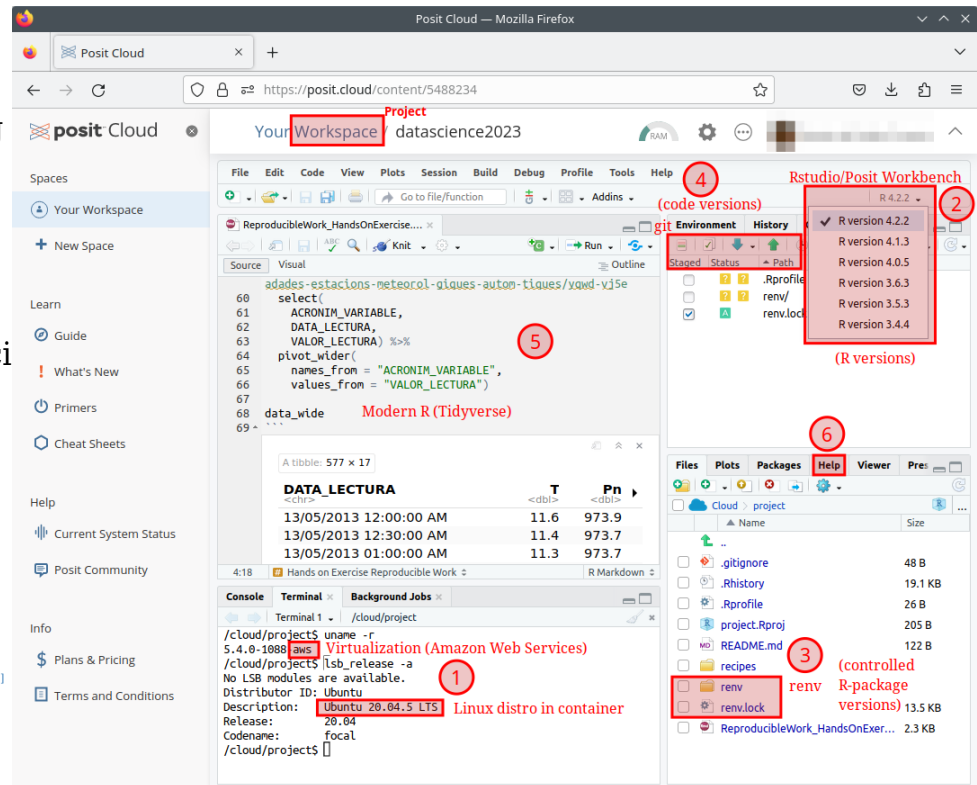


Thanks

Xavier de Pedro Punte,
Ph.D. -
xavier.depedro@seeds4c.org

Slides available at:
<https://seeds4c.org/reproduciblework2024>^[29]

[30]
Unless elsewhere noted, contents of this web site are released under a Creative Commons^[31] license.



[1] <https://www.il3.ub.edu>

[2] <https://seeds4c.org/reproduciblework2024>

[3] <https://stackoverflow.com/questions/30492623/using-both-python-2-x-and-python-3-x-in-ipynotebook>

[4] <https://posit.cloud>

[5] <https://kubernetes.io/docs/concepts/overview/>

[6] <https://rstudio.github.io/renv/articles/renv.html>

[7] <https://solutions.posit.co/envs-pkgs/environments/>

[8] <https://gitlab.com/radup/curs-r-introduccio/>

[9] <https://gitlab.com/radup/curs-r-introduccio/-/tree/master/codi>

- ^[10] <https://gitlab.com/radup/curs-r-introduccio/-/raw/master/codi/10.compartir.via.git.pdf>
- ^[11] <https://seeds4c.org/git>
- ^[12] <https://solutions.posit.co/envs-pkgs/environments/>
- ^[13] <https://www.youtube.com/watch?v=VjDM-XsoHUQ>
- ^[14] <https://www.youtube.com/watch?v=Cn-72tbRNfc&t=79s>
- ^[15] <https://github.com/data-and-visualization/git-tutorial>
- ^[16] <https://www.youtube.com/watch?v=GrqM2VqIQ20>
- ^[17] <https://www.youtube.com/watch?v=yjIEblDevOs>
- ^[18] <https://rstudio.github.io/renv/>
- ^[19] <https://github.com/workflowr/workflowr>
- ^[20] <https://rstudio.github.io/learnr/>
- ^[21] <https://roxygen2.r-lib.org/>
- ^[22] <https://www.tidyverse.org/>
- ^[23] <https://posit.cloud/plans/free>
- ^[24] <https://blog.djnavarro.net/using-aws-s3-in-r>
- ^[25] <https://doi.org/10.1371/journal.pbio.2006930>
- ^[26] <https://posit.cloud/plans/free>
- ^[27] <https://gitlab.com/xavidp/datascience2023>
- ^[28] <https://cran.r-project.org/src/contrib/Archive/packageName/>
- ^[29] <https://seeds4c.org/reproduciblework2024>
- ^[30] <http://creativecommons.org/licenses/by-sa/3.0/>
- ^[31] <http://creativecommons.org/licenses/by-sa/3.0/>