

# 2023 Reproducible work in Data Science (X. de Pedro)

"Data Science. Applications to Biology and Medicine with Python and R", at IL3 - University of Barcelona. Feb 27th, 2023 (16-19:15h).

Content at <https://seeds4c.org/reproduciblework2023>

The screenshot displays the Posit Cloud interface within a Mozilla Firefox browser window. The workspace is titled "Your Workspace / datascience2023". The main editor shows R code for data manipulation:

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

A preview window titled "A tibble: 577 x 17" displays a table with the following data:

DATA_LLECTURA	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 terminal window shows the following output:

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

The environment panel shows the R version 4.2.2 selected from a dropdown menu. The file explorer shows the project structure:

- .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)

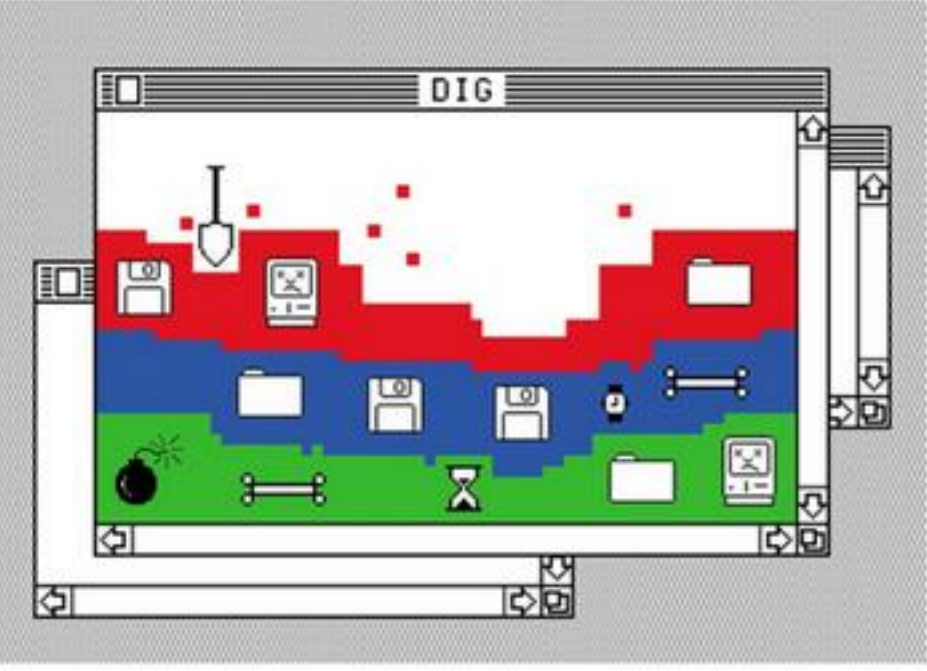
# 1. Introduction - the problems (i)

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.

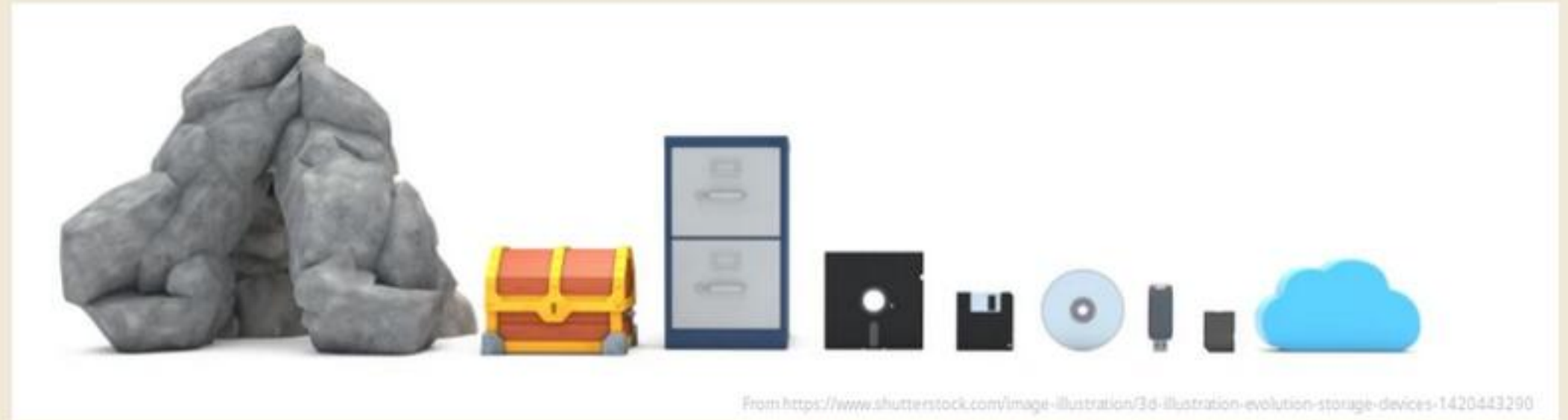
Jeffrey M. Perkel



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

Obsolete Devices storing code & data

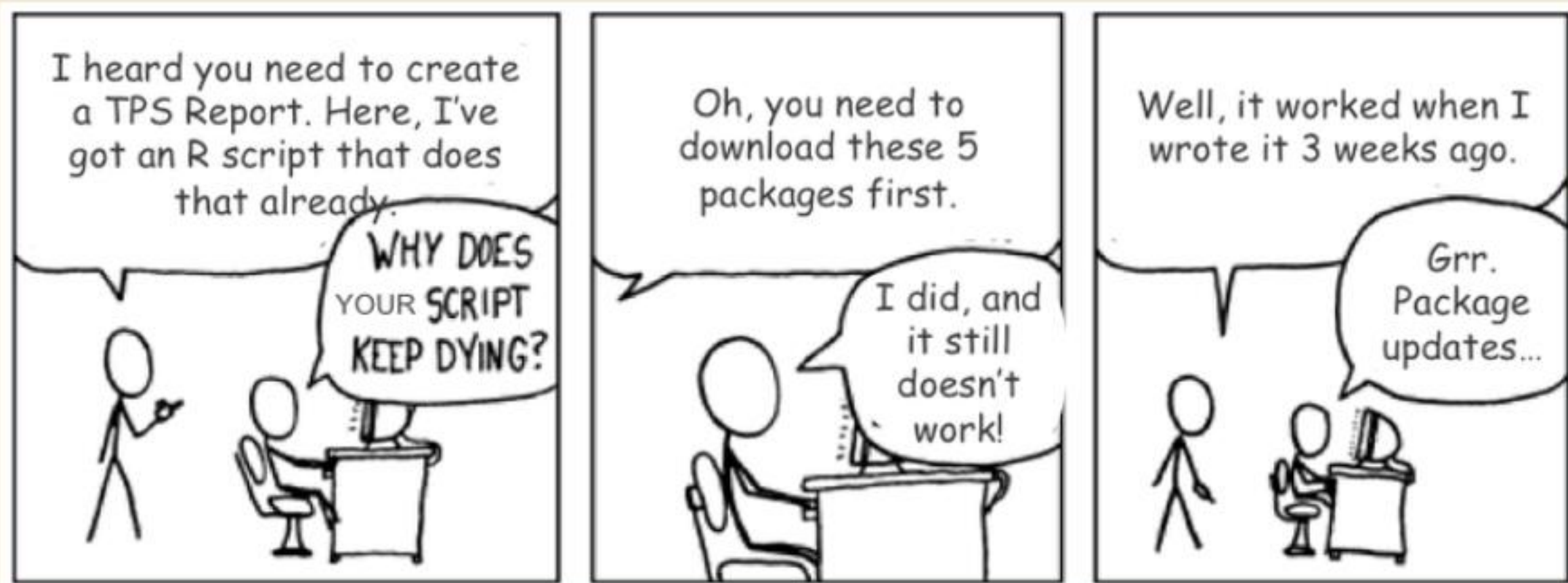
-->



-->

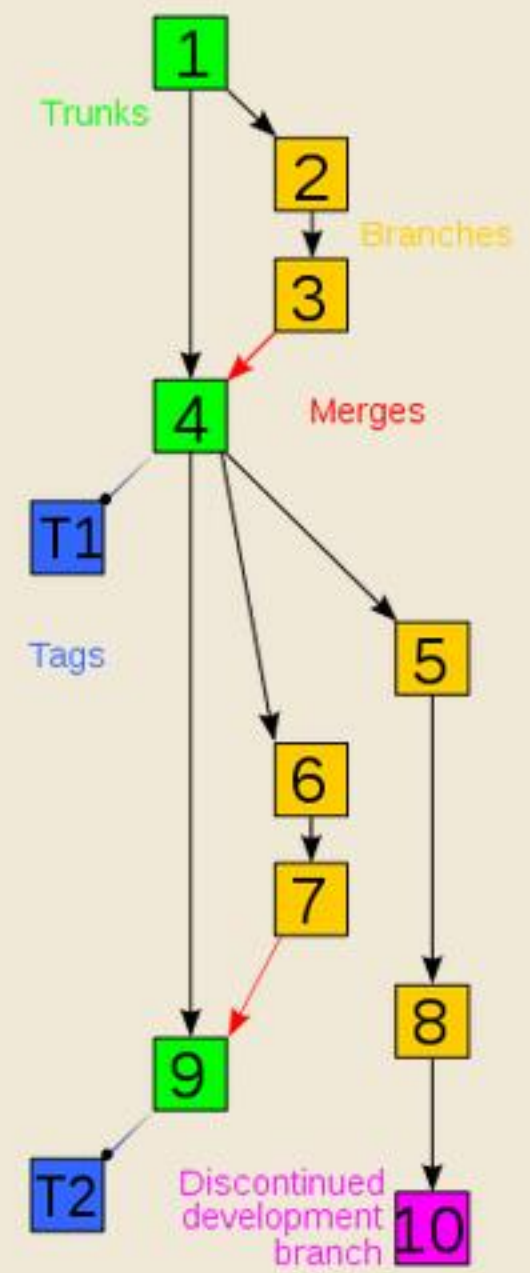
Ease copying to new devices (legally also: copyleft, ...) + online repositories

# 1.1. The problems (ii)



Software obsolescence and incompatible dependency versions

-->



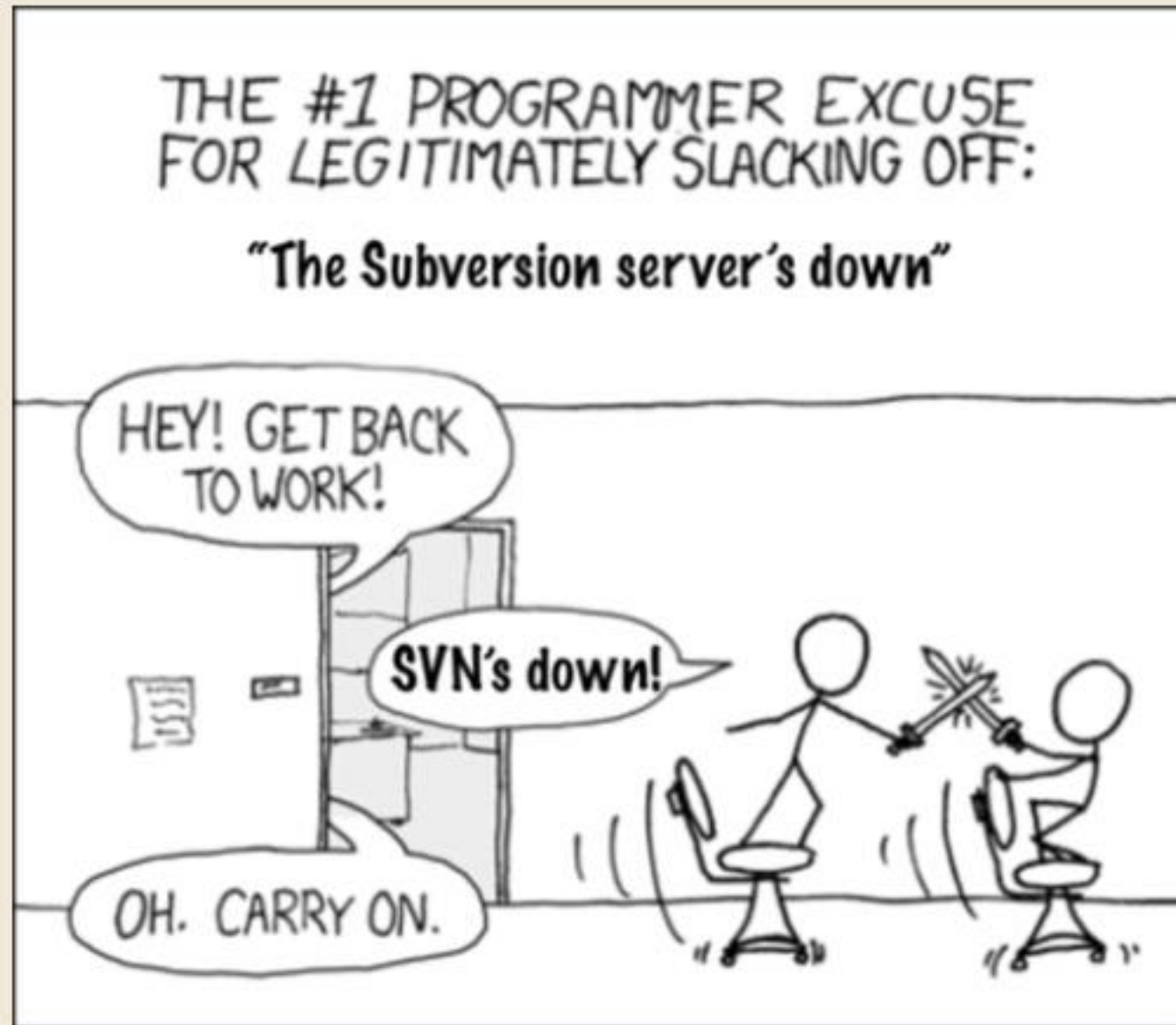
-->

Adapt to code evolution:

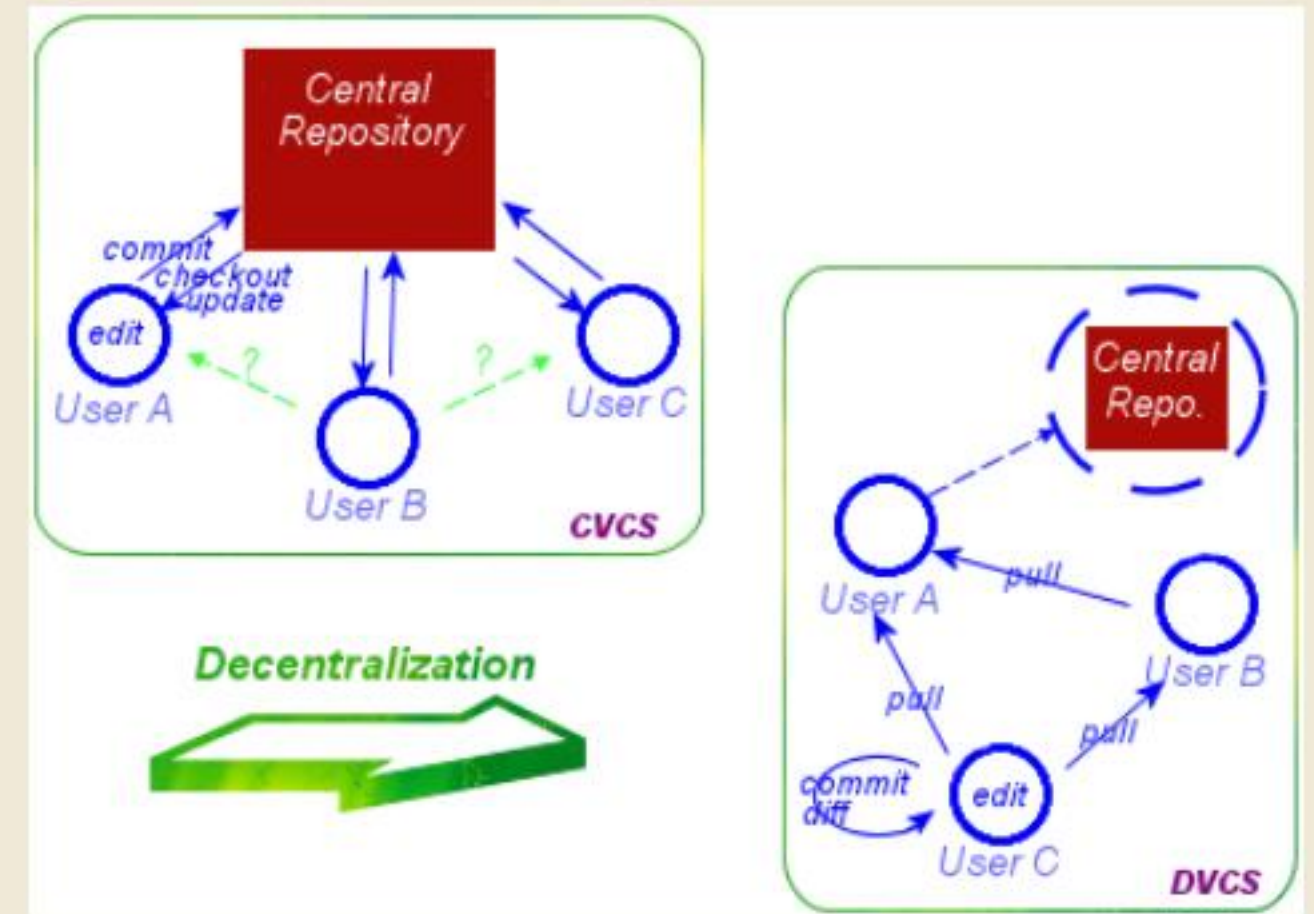
- Controlling Package Versions ( `renv` )
- VCS (`git`, `bazaar`, `svn`...)

VCS = Version Control Systems

# 1.2. The problems (iii)



-->

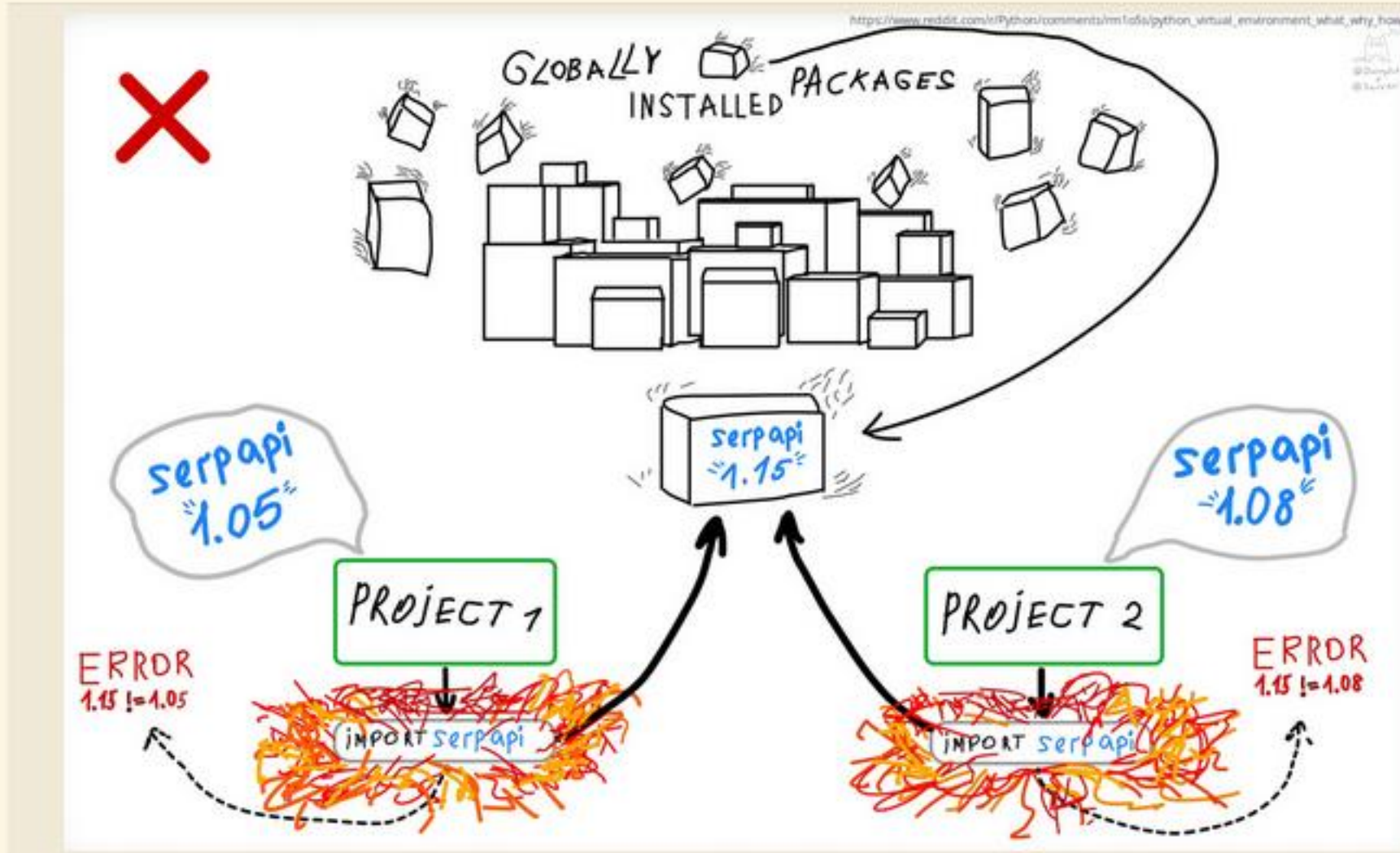


Centralization (such as Subversion VCS (svn) may increase efficiency but it also decreases Resilience ("shit happens")

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

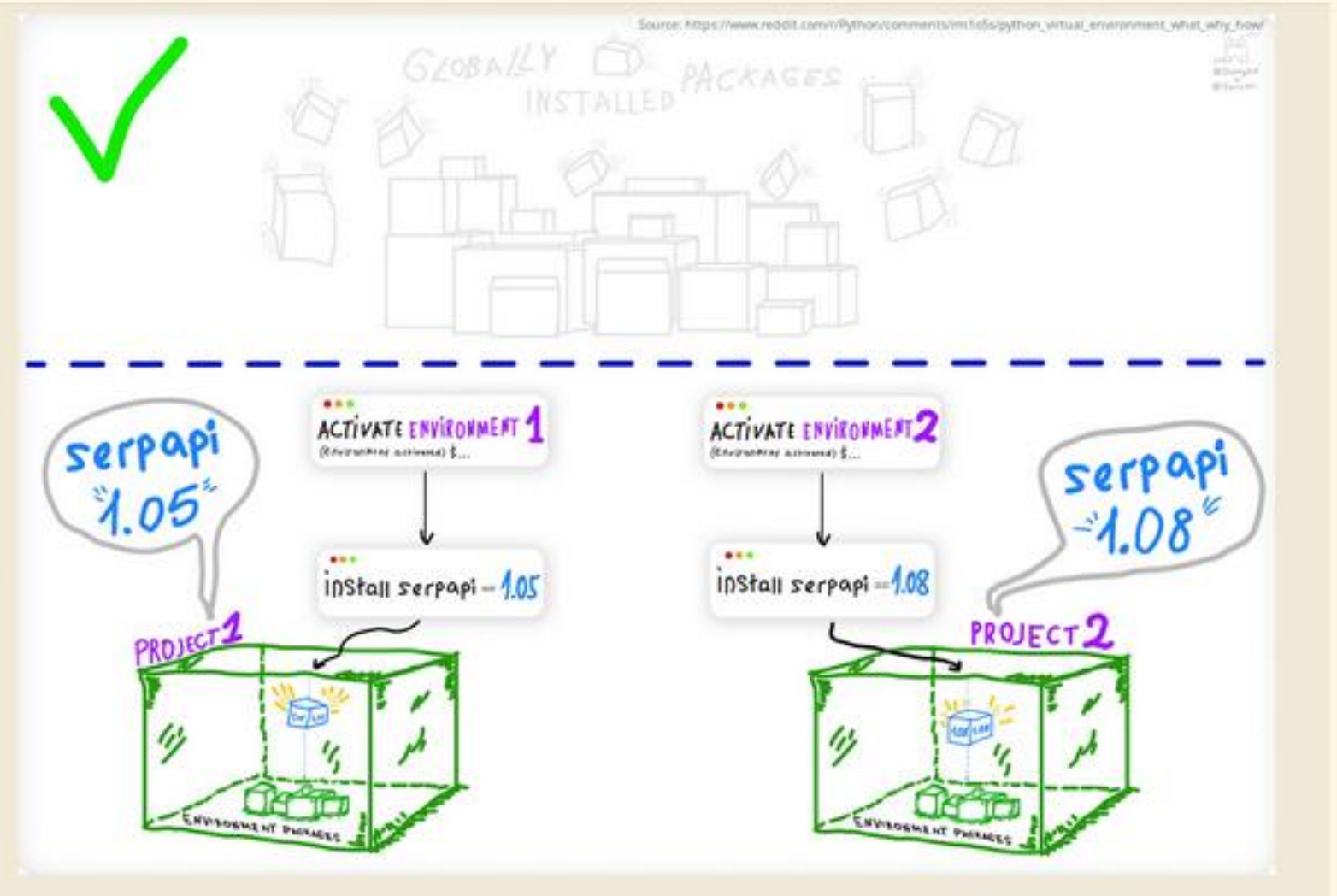
VCS = Version Control Systems

# 1.3. The problems (iv)



Conflicting package versions at system level with package versions at project levels

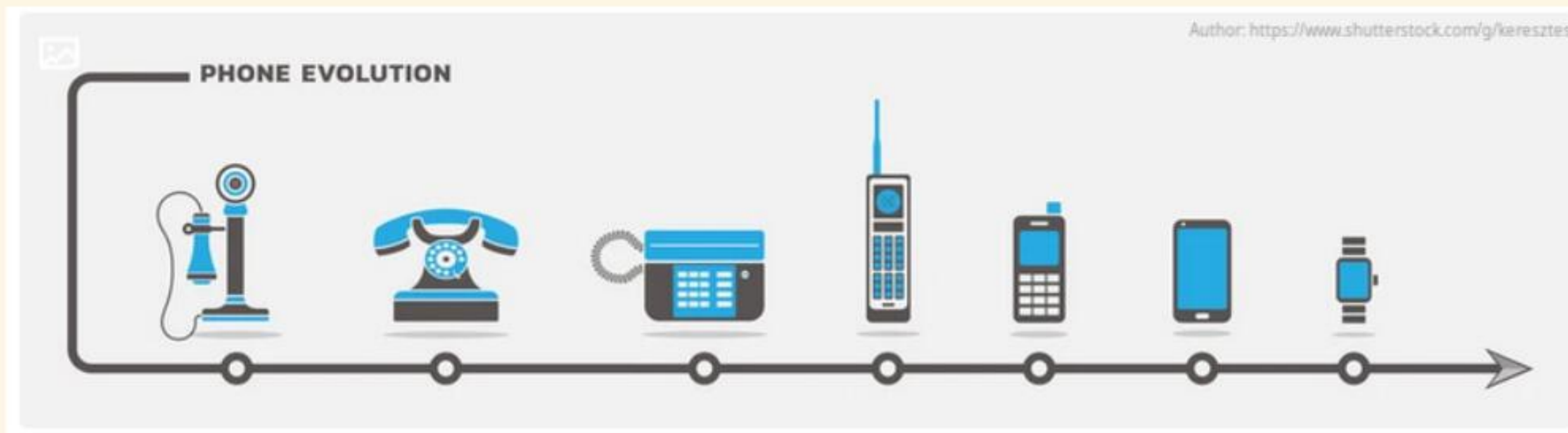
-->



Package versions per project Environment ( `renv` )

-->

## 1.4. The problem (v)



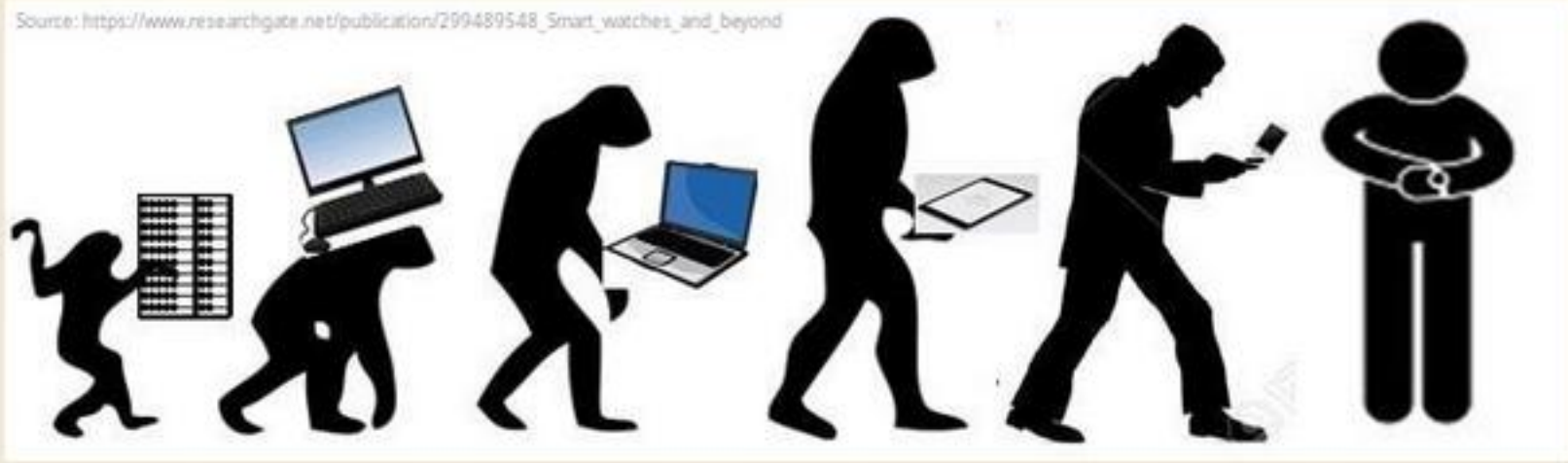
Sometimes a project was developed with a major version of a programming language (R 3.x, Python 2.x), while another project in the same server requires a different major version (R 4.x, Python 3.x)

--> R case: from RStudio Server to Posit Workbench (former *RStudio Server Pro*)

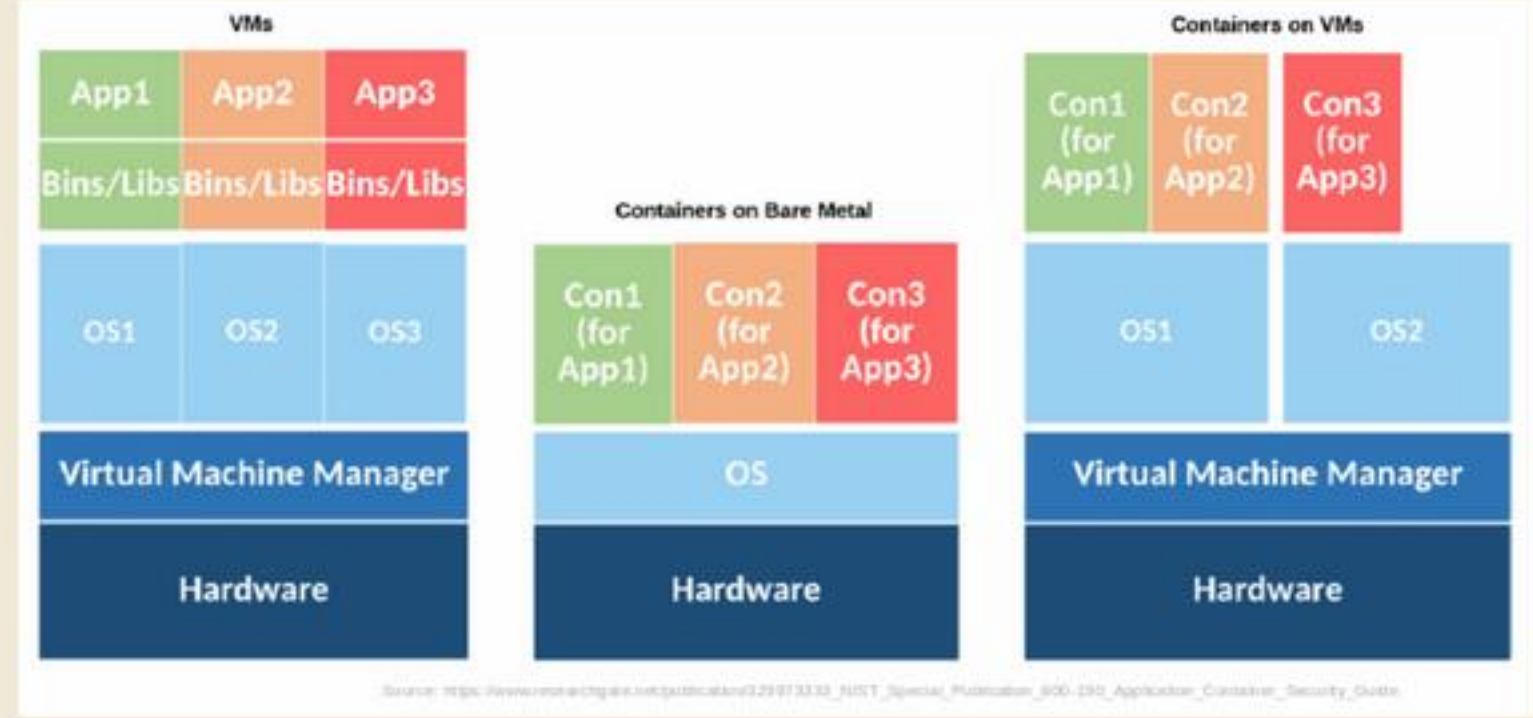
You can choose R version per project

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

# 1.5. The problem (vi)



-->



Changes per Operating System itself (32 bit systems unsupported anymore, discontinued linux distros, ...)

-->

**Virtual Machines or Containers (OVA, KVM, LXC, Docker, Pod...):** you can choose OS version per container within the same server



## 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** (algorithm 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):

ISSUES	SOLUTIONS / WORKAROUNDS
(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, Google Colab, ... 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, ...), ...  VCS = <i>Version Control system</i>
(Level 5) Tidy script content and organization	<u>Literate Coding</u> (Scripting & Coding) / Analysis  - R: <u>Rstudio Notebooks</u> with modern R ( <i>Tidyverse</i> ). VS Notebooks, G-Colab, ... - Python: <u>Jupyter Notebooks</u> , 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> (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** , ...)

The screenshot displays the Posit Cloud interface in a Mozilla Firefox browser. The main workspace is titled 'Your Workspace' and 'Project' 'datascience2023'. The interface is annotated with red circles and boxes highlighting key features:

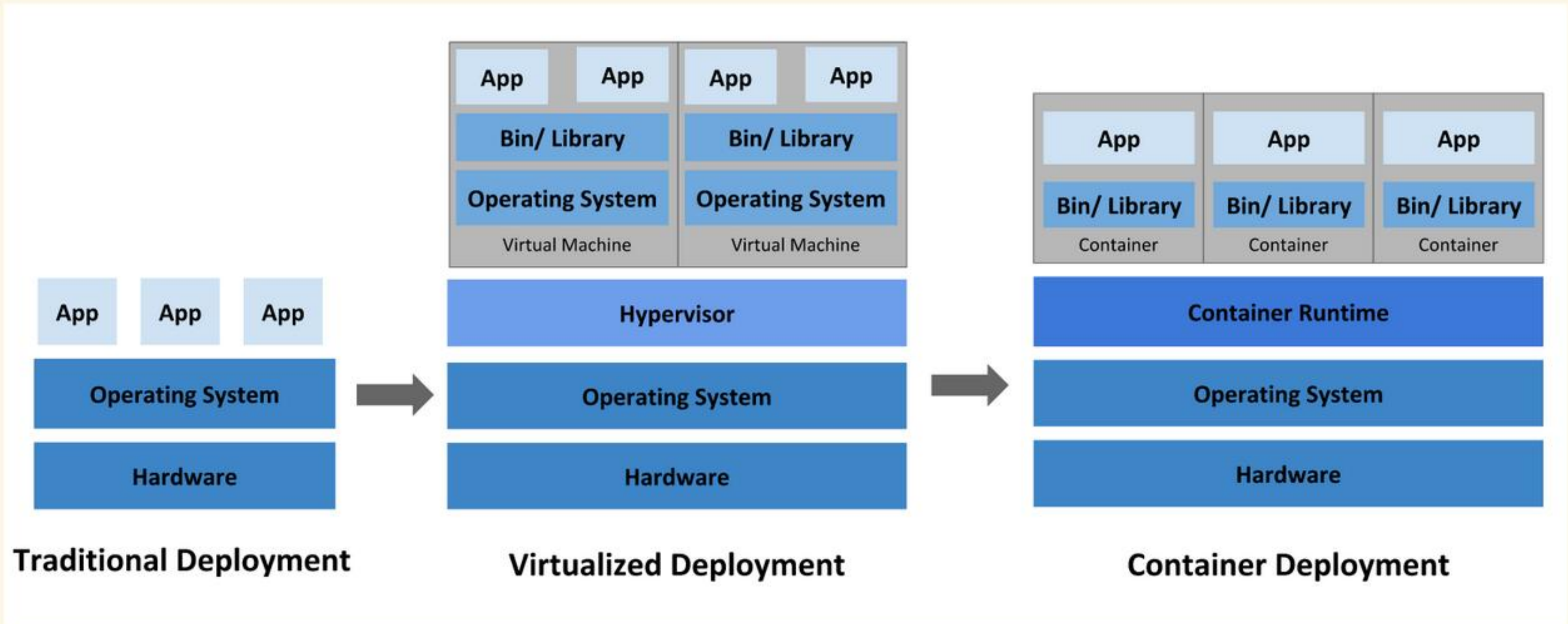
- 1**: Terminal output showing 'Virtualization (Amazon Web Services)' and 'Linux distro in container' (Ubuntu 20.04.5 LTS).
- 2**: R version selection dropdown menu showing options from R 4.2.2 down to R 3.4.4.
- 3**: File browser showing the 'renv' directory and 'renv.lock' file, labeled as '(controlled R-package versions) 13.5 KB'.
- 4**: RStudio/Posit Workbench toolbar, labeled as '(code versions)'.
- 5**: R code editor showing a script using 'Modern R (Tidyverse)' syntax.
- 6**: The 'Help' menu in the RStudio interface.

The console output at the bottom shows the following system information:

```
/cloud/projects$ uname -r
5.4.0-1888-aws Virtualization (Amazon Web Services)
/cloud/projects$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.5 LTS Linux distro in container
Release:        20.04
Codename:       focal
/cloud/projects$
```

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

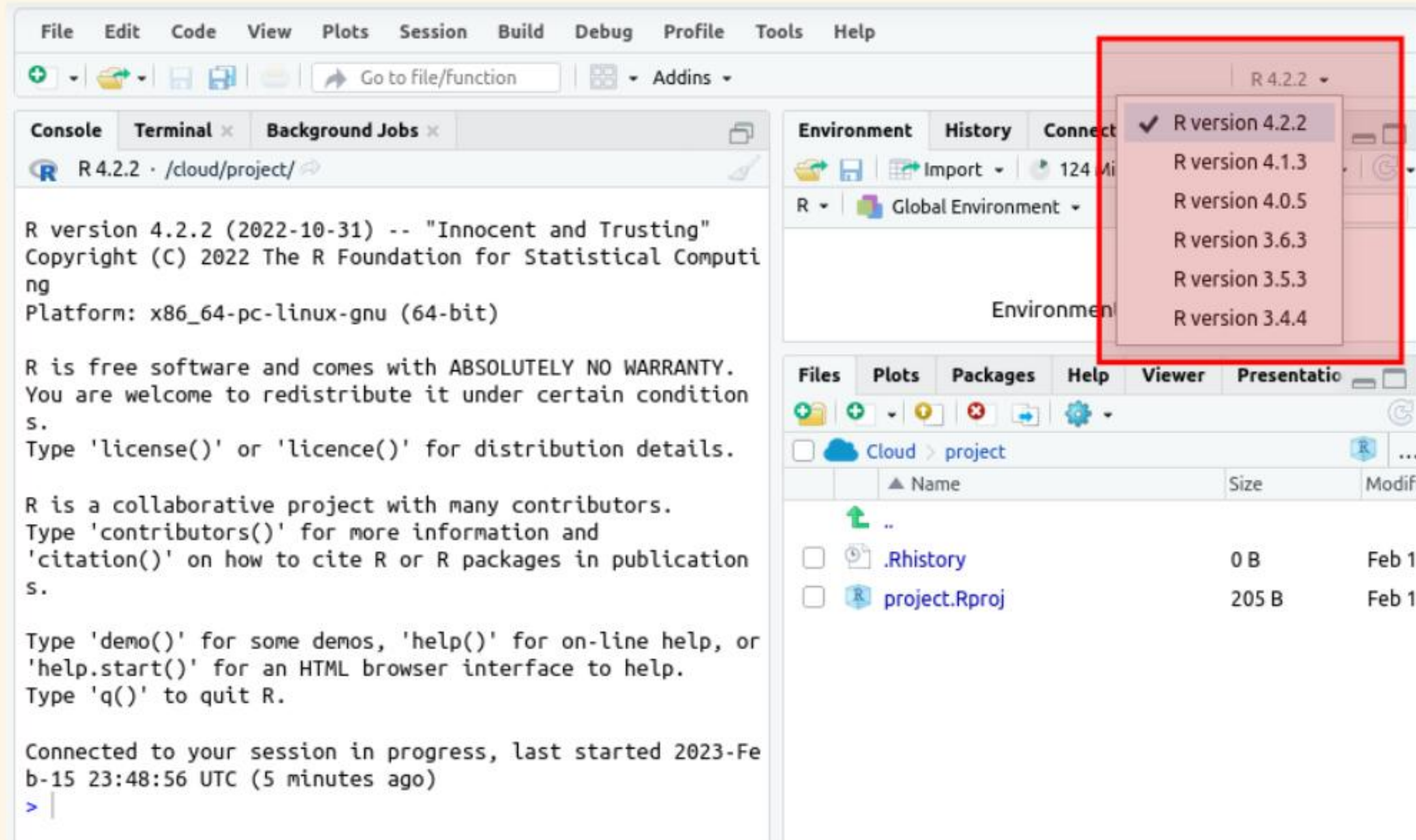
# 4.1. Level 1: Virtual Machines or Containers



From:

<https://kubernetes.io/docs/concepts/overview/>

## 4.2. Level 2: RStudio-Posit Workbench



The screenshot displays the RStudio-Posit Workbench interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu bar is a toolbar with icons for file operations and a search bar labeled "Go to file/function". The main workspace is divided into several panes:

- Console:** Shows the R version 4.2.2 (2022-10-31) -- "Innocent and Trusting" and the platform x86\_64-pc-linux-gnu (64-bit). It also displays the R license and contributors information.
- Environment:** Shows the Global Environment.
- Files:** Shows a file browser with a table of files in the "project" directory.

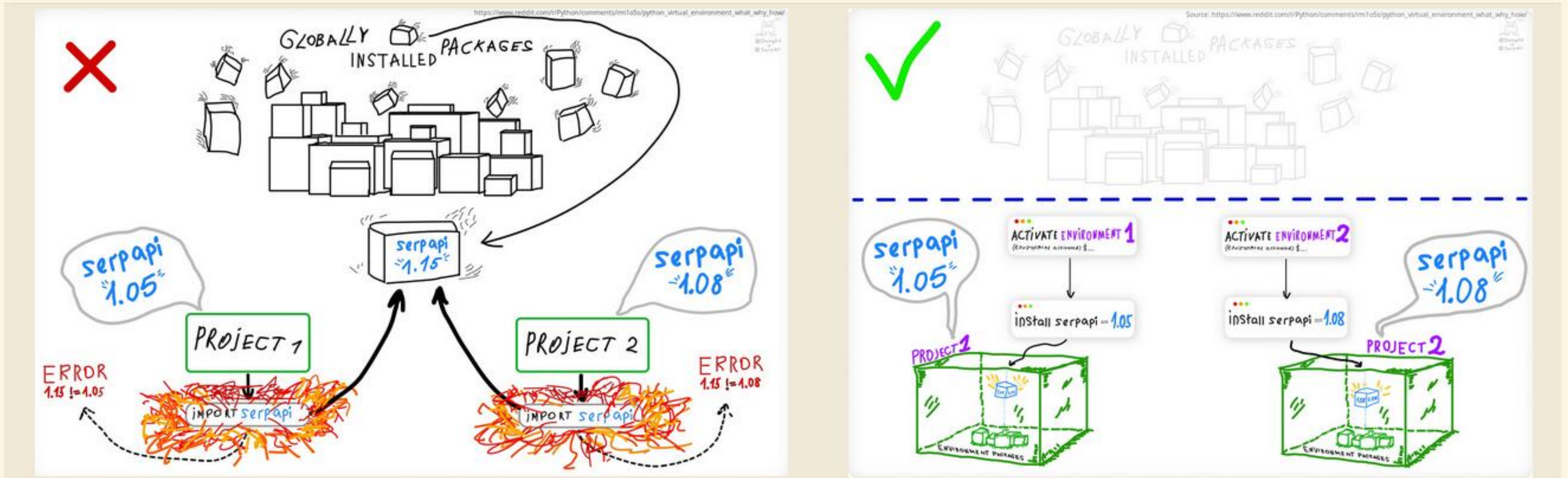
A red box highlights the R version selection menu, which is currently set to R 4.2.2. The menu lists the following options:

- R version 4.2.2
- R version 4.1.3
- R version 4.0.5
- R version 3.6.3
- R version 3.5.3
- R version 3.4.4

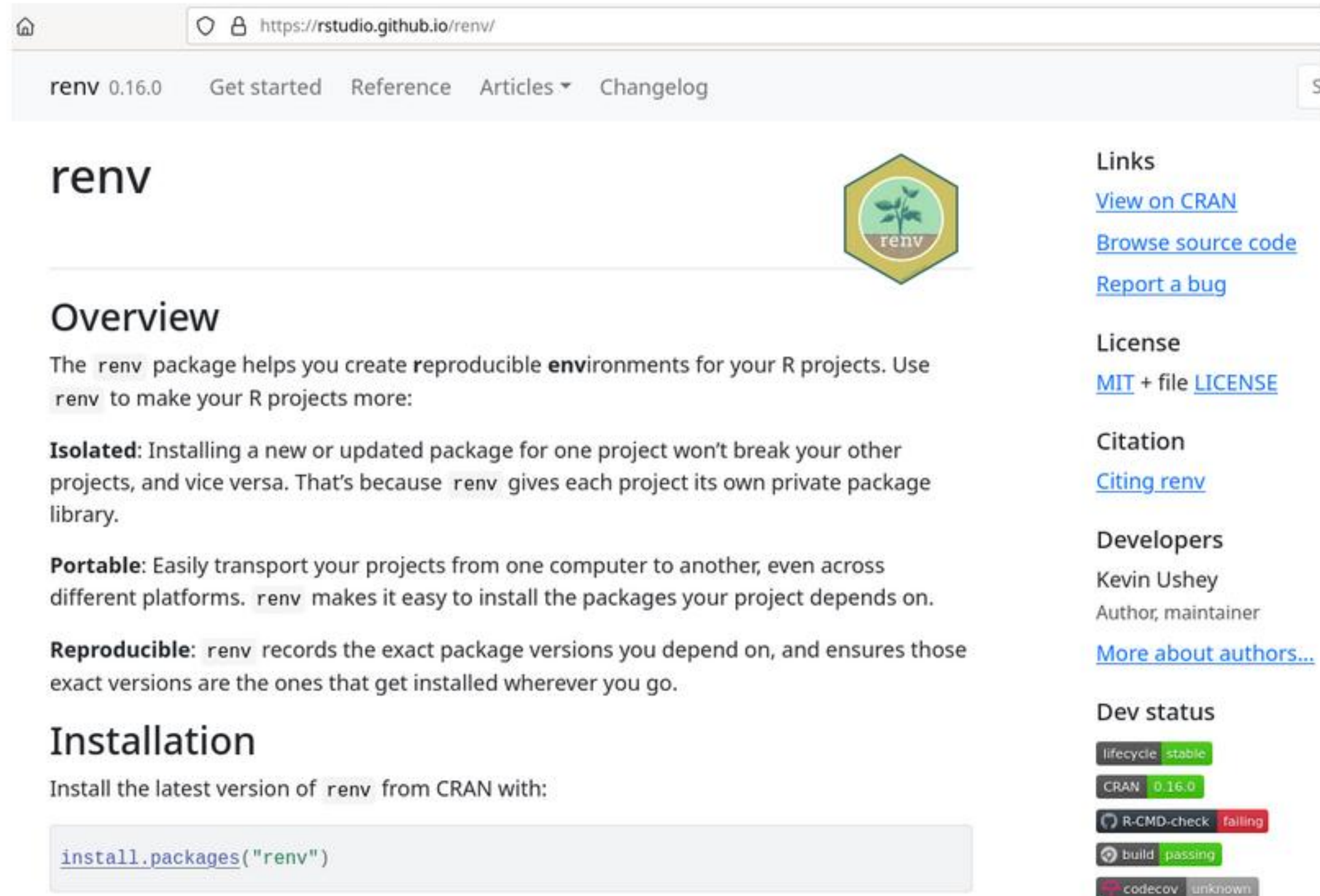
Name	Size	Modified
..		
.Rhistory	0 B	Feb 1
project.Rproj	205 B	Feb 1

# 4.3. Level 3: renv - for packages

Version control in work "environments"



# 4.3.1. Virtual environments in R with renv



The screenshot shows the GitHub page for the `renv` package. The browser address bar shows `https://rstudio.github.io/renv/`. The page header includes the package name `renv` 0.16.0 and navigation links: `Get started`, `Reference`, `Articles`, and `Changelog`. The main content area features the `renv` logo (a green hexagon with a plant) and an **Overview** section. The overview text states: "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." Below the overview is an **Installation** section with the instruction "Install the latest version of `renv` from CRAN with:" and a code block containing `install.packages("renv")`. On the right side, there are sections for **Links** (with links to `View on CRAN`, `Browse source code`, and `Report a bug`), **License** (with links to `MIT` and `LICENSE`), **Citation** (with link to `Citing renv`), **Developers** (listing Kevin Ushey as author/maintainer and a link to `More about authors...`), and **Dev status** (showing lifecycle as `stable`, CRAN as `0.16.0`, R-CMD-check as `failing`, build as `passing`, and codecov as `unknown`).

## 4.3.2. From utils::sessionInfo() to renv::snapshot() + renv.lockalso 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_1.0.10
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
```

renv::snapshot() | 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": {
```

## 4.3.3. "Happy path"

For a reproducible environment

Commands in terminal - Computer 1

```
1 cd project_folder
2 git init
3 R
4 [ouvrir projecte de RStudio]
5 renv::init() # to initialize renv in your code project
6 renv::snapshot() # to make a snapshot "picture" of the list of R packages used within the whole R project and their respective package versions
7 q()
8 git commit ...
9 git push
```

Commands in terminal - Computer 2

```
1 cd project_folder
2 git clone/git pull ...
3 R
4 [open same RStudio project]
5 renv::status() # for a report on which steps are suggested for you to follow
6 renv::restore() # to restore the package library (with the required package versions) for this project
7 [continue working in/developing your code]
8 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 ;-))
9 q()
10 git commit ...
11 git push
```



## 4.3.4. Infrastructure

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 <code>.Rprofile</code> .
<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

1 **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/>

2 **renv::record("packagename", version="0.1")** # to save at `renv.lock` the specific version you need for this package

3 **renv::deactivate()** # to temporarily deactivate `renv` in your project

4 **renv::activate()** # to reactivate `renv` in your project

5 **renv::equip()** # for special installations in MS Windows

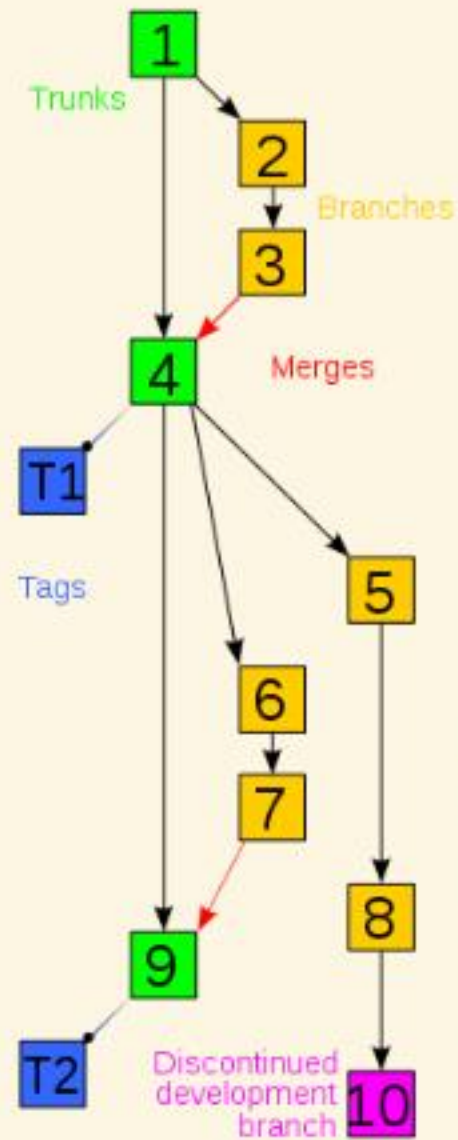
6 **vignette("docker", package = "renv")** # for a combined use with Docker

7 **vignette("collaborating", package = "renv")** # to improve collaborative use in work teams

And much more. See:

- <https://rstudio.github.io/renv/articles/renv.html>
- <https://solutions.posit.co/envs-pkgs/environments/>

# 4.4. Level 4: git - for code



RStudio: Review Changes

Changes History master Stage Revert Ignore Refresh Pull Push

Staged	Status	Path
<input type="checkbox"/>	U U	.gitignore
<input checked="" type="checkbox"/>	M	my.R
<input type="checkbox"/>	U U	r.Rproj

Commit message: My important changes from today

Amend previous commit

Show  Staged  Unstaged Context 5 line Unstage All

```
@@ -1,9 +1,10 @@
1 1 # My.R script
2 2
3 3 ##### Chunk 1: foo #####
4 4 # -----#
5 5 print("foo")
6 6 cat("foo")
7 7 date()
8 8
9 9 ##### Chunk 2: bar #####
10 10 ##### Chunk 2: bar2 #####
11 11 # -----#
12 12 print("bar")
13 13 No newline at end of file
14 14 print("bar2")
15 15 No newline at end of file
```

RStudio: Review Changes

Git Commit [Close]

[master 0453e49] My important changes from today  
1 file changed, 5 insertions(+), 4 deletions(-)

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

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

# 5. More information

## Work Environments in R

---

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

## Videos

---

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

## R Packages

---

[renv](#) | [workflowr](#) | [learnr](#) | [roxygen2](#) | [Tidyverse](#)

## Free Work environments for Collaborative Data Science with R & Python

---

- <https://posit.cloud/plans/free>

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

---

- Danielle Navarro. 2022. "[Using Amazon S3 with R](#)" 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>
- 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. Hand on practical exercise

The screenshot displays the Posit Cloud interface within a Mozilla Firefox browser window. The browser address bar shows the URL `https://posit.cloud/content/5488234`. The main workspace area is titled "Your Workspace / datascience2023".

The interface includes a sidebar on the left with navigation options: Spaces (Your Workspace, New Space), Learn (Guide, What's New, Primers, Cheat Sheets), Help (Current System Status, Posit Community), and Info (Plans & Pricing, Terms and Conditions).

The central workspace contains a code editor with the following R code:

```
adades-estacions-meteorol-giques-auton-tiques/yawd-vj5e
60 select(
61   ACRONIM_VARIABLE,
62   DATA_LLECTURA,
63   VALOR_LLECTURA) %>%
64   pivot_wider(
65     names_from = "ACRONIM_VARIABLE",
66     values_from = "VALOR_LLECTURA")
67
68 data_wide
69 ...
```

Below the code editor, a data preview window shows a tibble with 577 rows and 17 columns. The visible columns are `DATA_LLECTURA` (character), `T` (double), and `Pn` (double). The data rows are:

DATA_LLECTURA	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 bottom section of the interface features a console window with the following terminal output:

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

On the right side, there is an "Environment" panel showing a dropdown menu for R versions with "R version 4.2.2" selected. Below it is a file explorer showing the project structure:

Name	Size
..	
.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

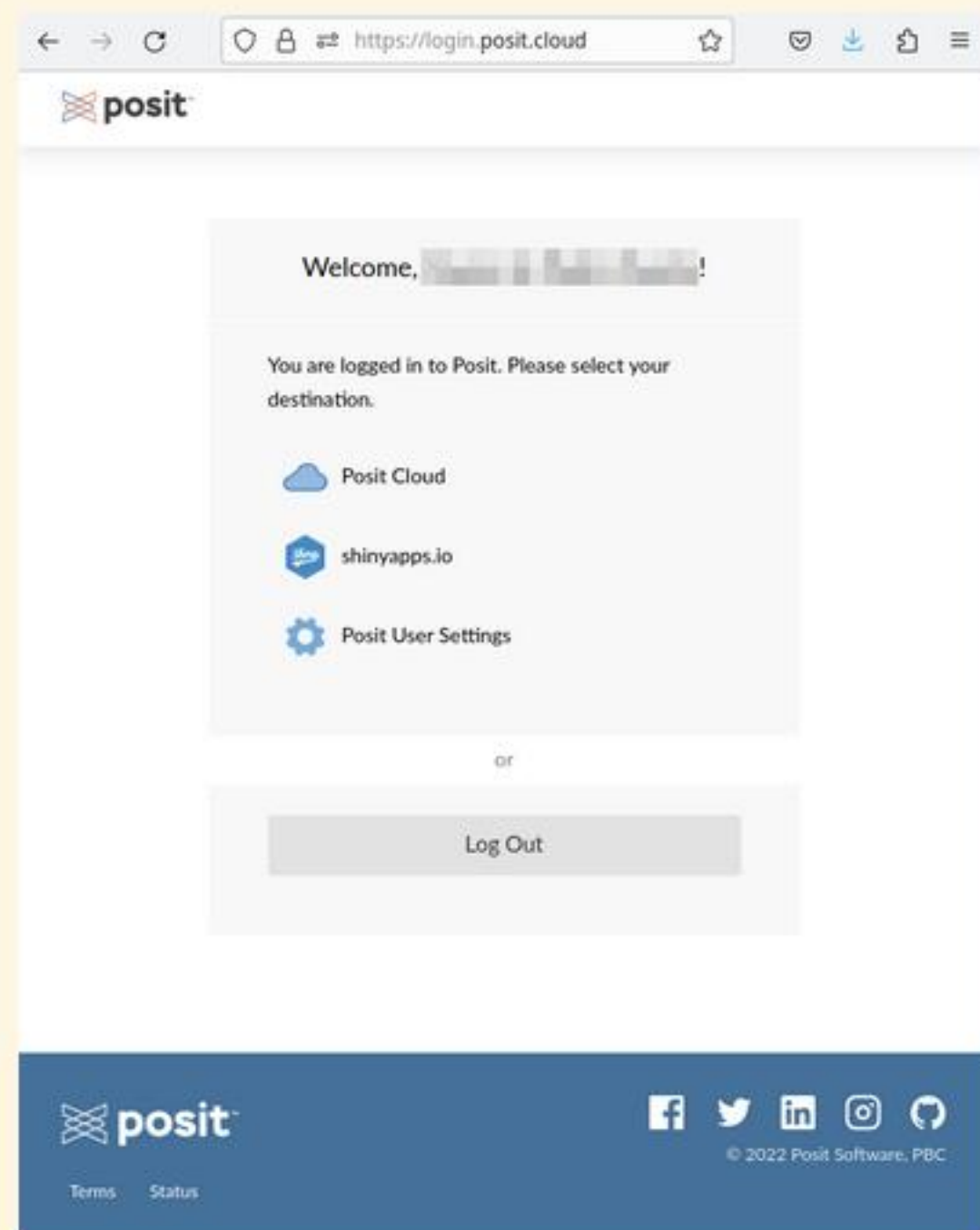
## 6.1. Register a free account a Posit Cloud

You can do so at:

- <https://posit.cloud/plans/free>

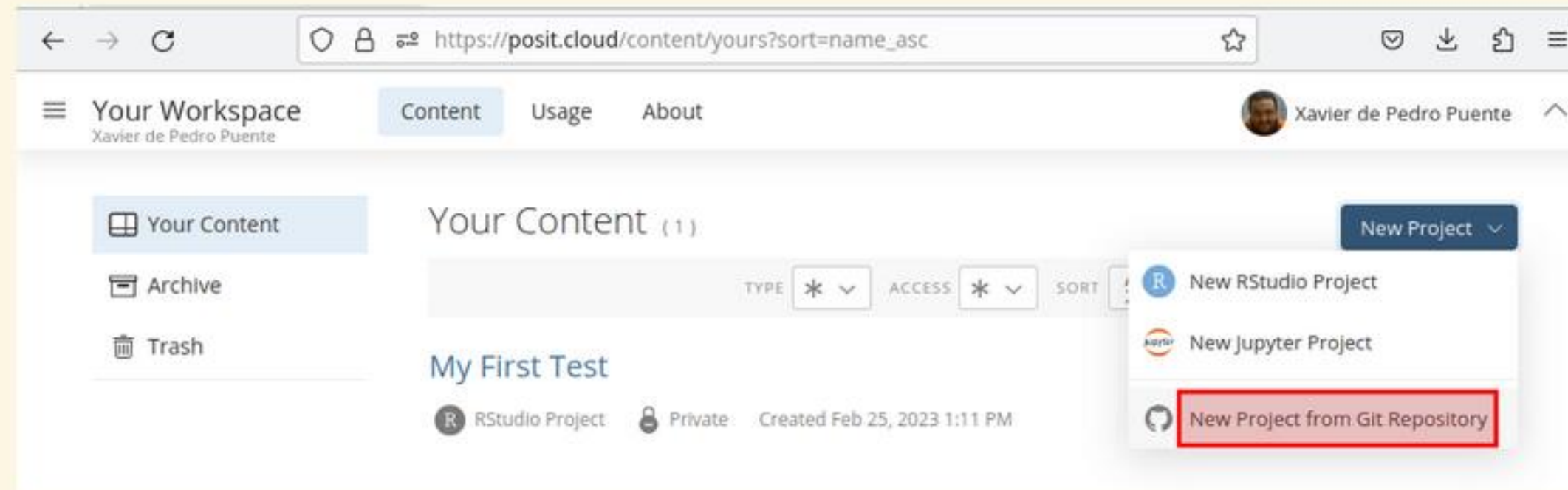
You will need to click on a link sent to your email inbox to validate your account.

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>

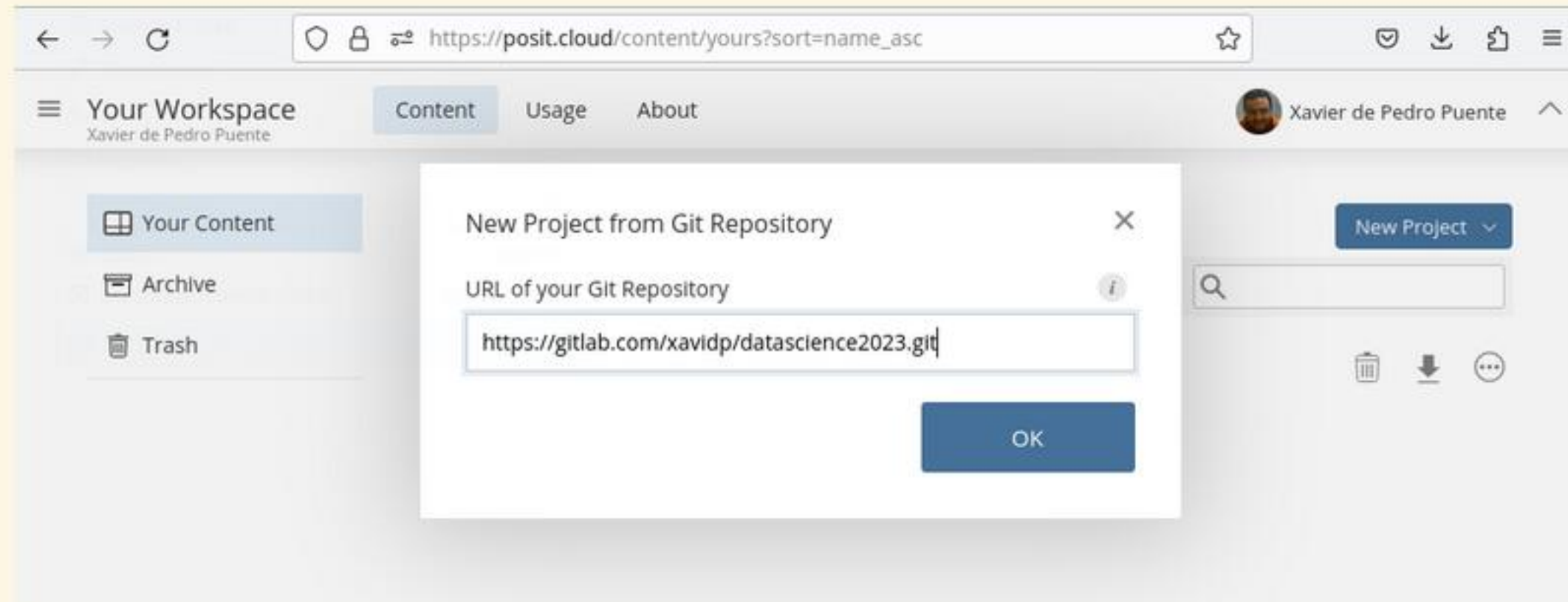
The screenshot shows a web browser displaying the GitLab repository page for 'DataScience2023'. The URL in the address bar is `https://gitlab.com/xavidp/datascience2023`. The repository page includes a sidebar with navigation icons, a search bar, and a header with the repository name and project ID. Below the header, there are statistics for commits, branches, tags, and project storage. A recent commit titled 'Base Rmd file' is shown. The main content area features a 'Clone' button, which is highlighted with a red box. A dropdown menu is open, showing options for cloning with SSH and HTTPS. The HTTPS option is also highlighted with a red box. Below the dropdown, there are options to 'Open in your IDE' using Visual Studio Code or IntelliJ IDEA via SSH or HTTPS. A table of files and their last commit messages is visible at the bottom of the page.

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:



# 6.3. Choose R 3.6.x & Run Rmd

The screenshot shows the Posit Cloud interface in a Mozilla Firefox browser. The browser address bar shows `https://posit.cloud/content/5488234`. The main workspace area is titled "Your Workspace / datascience2023".

The interface includes a menu bar (File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help) and a toolbar with icons for file operations and a "Go to file/function" search bar. The "Addins" dropdown is also visible.

The main editor displays a source file named "ReproducibleWork\_HandsOnExercise.Rmd". The code in the editor includes:

```
1 ---  
2 title: "Hands on Exercise Reproducible Work"  
3 author: "Xavier"  
4 date: "2023-02-25"  
5 output: html_document  
6 ---  
7  
8 {r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10  
11  
12 # Session Reproducible Work  
13  
14 Monday Feb 27, 2023. IL3-UB.
```

The "Run" button in the toolbar is highlighted with a red circle (1). A red arrow points from this button to the "Run All" option in the dropdown menu, which is also highlighted with a red circle (2). The "Run All" option has the keyboard shortcut "Ctrl+Alt+R" and a tooltip that says "Run all of the code in the source file".

The "Environment" panel on the right shows "R 3.6.3" selected (highlighted with a red circle (3)) and "Global Environment" as the current environment. The environment is currently empty.

The "Files" panel at the bottom right shows a file tree for the "project" directory. The file "ReproducibleWork\_HandsOnExercise.Rmd" is highlighted with a red circle (4).

The console at the bottom shows the R prompt and the following output:

```
R 3.6.3 · /cloud/project/  
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.  
  
[Workspace loaded from /cloud/project/.RData]  
> |
```

## 6.3.1. Install dependencies also

The screenshot shows the RStudio interface. The top-left pane displays the source code of an R Markdown document. A yellow warning banner at the top of the source pane reads: "Packages rmarkdown and knitr required but are not installed. **Install** Don't Show Again". The code in the source pane includes a YAML header with title, author, date, and output, followed by R code for session setup and a session title. The top-right pane shows the Environment tab, which is currently empty, displaying "Environment is empty". The bottom-right pane shows the Files tab with a file browser view.

```
1 ---
2 title: "Hands on Exercise Reproducible Work"
3 author: "Xavier"
4 date: "2023-02-25"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 # Session Reproducible Work
13
14 Monday Feb 27, 2023. IL3-UB.
15
```

The screenshot shows the RStudio console output. The top part of the console shows the same R Markdown code as the previous screenshot. Below the code, the console displays the output of the R package installation process. The output includes the following text:

```
11
12 # Session Reproducible Work
13
14 Monday Feb 27, 2023. IL3-UB.
15
16 Related to:
17 https://seeds4c.org/reproduciblework2023
18
19 4:5 Hands on Exercise Reproducible Work R Markdown
```

The console also shows the progress of installing R packages:

```
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 downloading and processing a dataset. Lines 27-29 are highlighted with a red box:

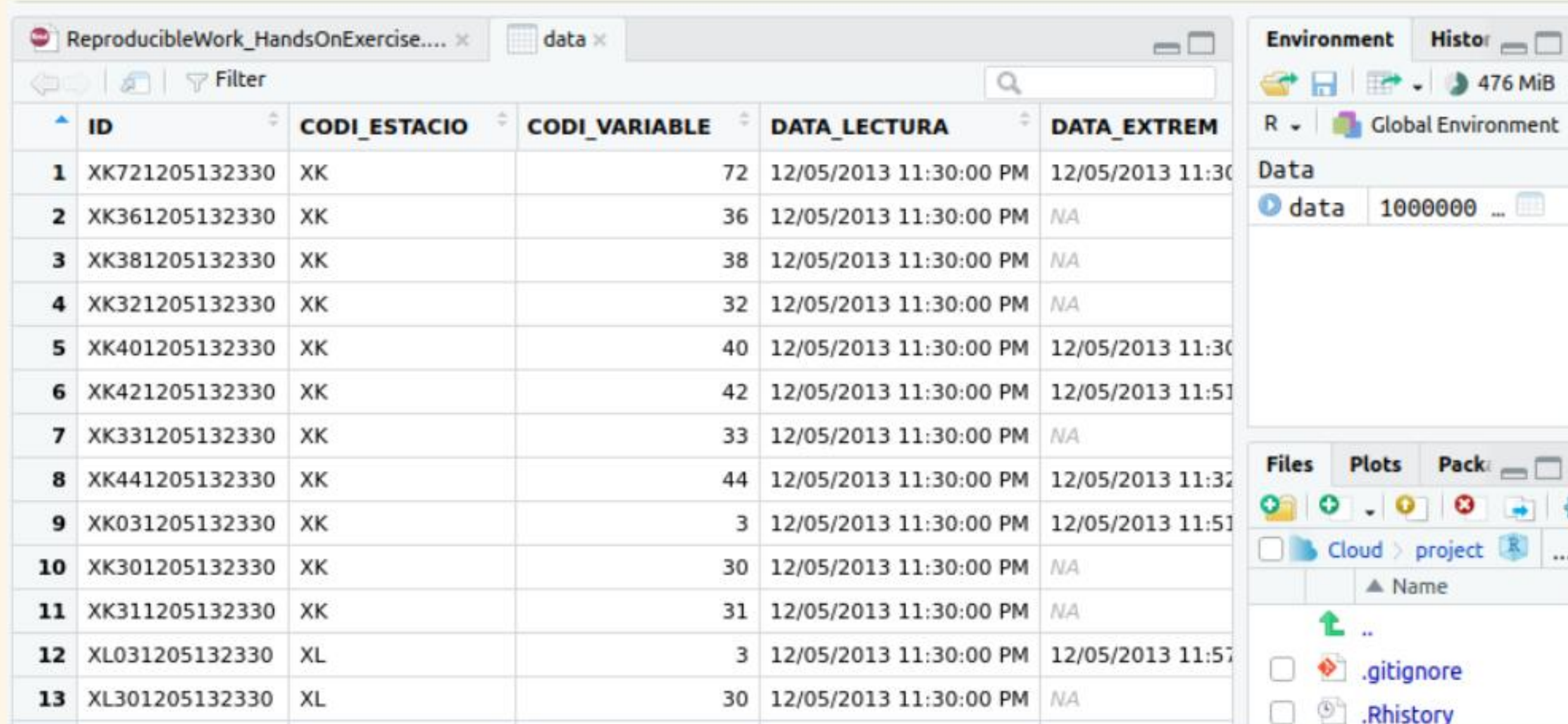
```
27 system("wget http://cloud.seeds4c.org/data_smc.csv.bz2")
28 system("bunzip2 data_smc.csv.bz2 -k")
29 system("cat data_smc.csv | head -n1000001 > data_subset.csv")
```
- Console:** Shows the execution of the R code, resulting in a data object with 1,000,000 rows and 8 columns.

```
> data <- read_csv("data_subset.csv")
Rows: 1000000 Columns: 8 Column specification
```
- Files Panel:** Shows the project directory structure. Files are listed with their sizes:

Name	Size
..	
.gitignore	48 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
- Environment Panel:** Shows the R environment with 535 MiB of memory used.
- Terminal:** Shows the R shell prompt and the execution of the R code.

## 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 ~ ```
```

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 ~ ```
```

A tibble: 26 × 2

CODI_VARIABLE	ACRONIM
<dbl>	<chr>
1	Px
2	Pn
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-autom-tiques/yqwd-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 - ```

data_wide
70 - ```

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

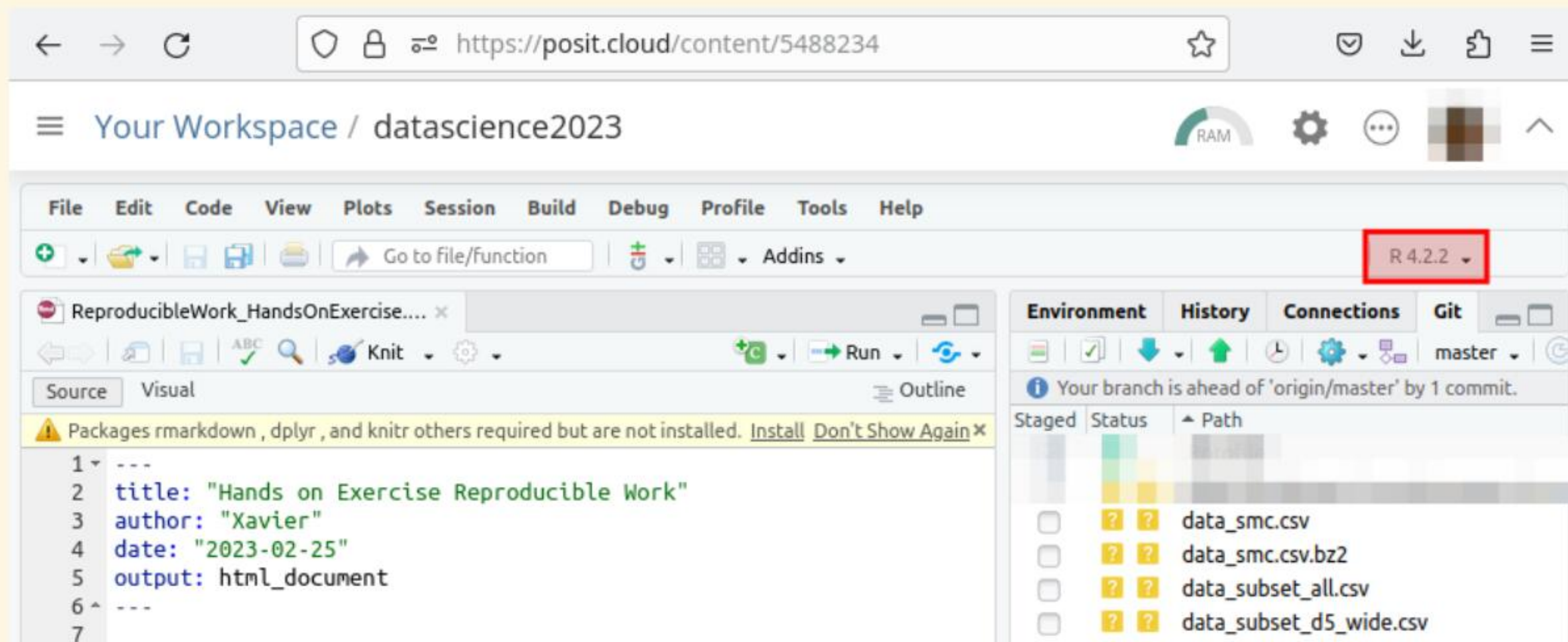
4:18  Hands on Exercise Reproducible Work ↕



## 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.



The screenshot shows the Posit Cloud workspace interface for 'datascience2023'. The browser address bar shows 'https://posit.cloud/content/5488234'. The workspace name is 'Your Workspace / datascience2023'. The menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar shows icons for file operations and a dropdown menu for the R version, which is currently set to 'R 4.2.2' and is highlighted with a red box. The main editor area shows a file named 'ReproducibleWork\_HandsOnExercise....' with a toolbar for Knit, Run, and other actions. A warning message at the top of the editor states: 'Packages rmarkdown, dplyr, and knitr others required but are not installed. Install Don't Show Again'. The code in the editor is:

```
1 ---
2 title: "Hands on Exercise Reproducible Work"
3 author: "Xavier"
4 date: "2023-02-25"
5 output: html_document
6 ---
7
```

The right sidebar shows the Environment, History, Connections, and Git panels. The Git panel indicates 'Your branch is ahead of 'origin/master' by 1 commit.' and lists several files in the Staged area:

Staged	Status	Path
<input type="checkbox"/>	??	data_smc.csv
<input type="checkbox"/>	??	data_smc.csv.bz2
<input type="checkbox"/>	??	data_subset_all.csv
<input type="checkbox"/>	??	data_subset_d5_wide.csv

## 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/>
- `renv::record("packagename@x.y.z")`
- `renv::snapshot()` # after all packages installed without any more issues

```
Console Terminal x Background Jobs x
R 3 - /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:

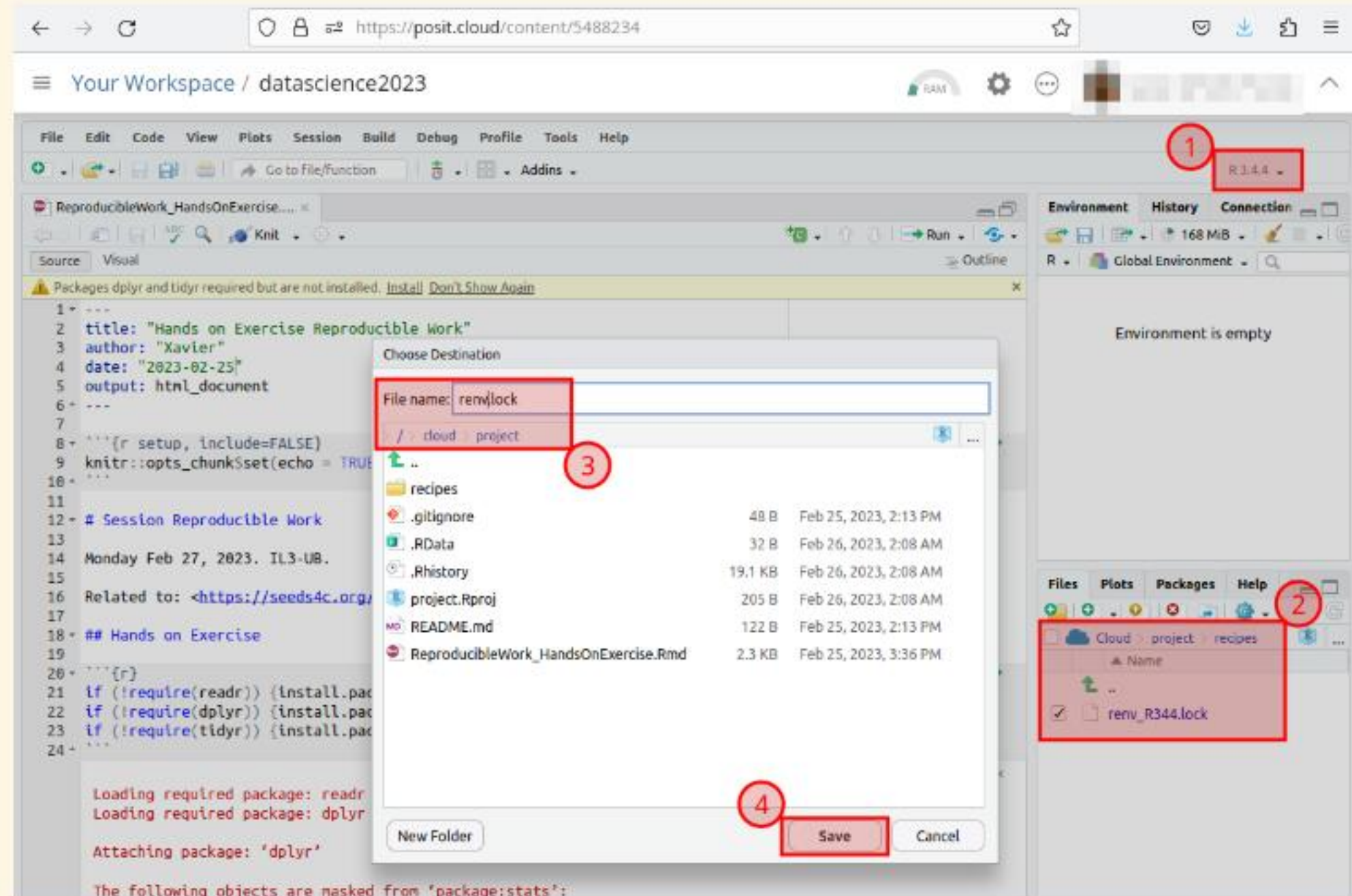
# RSPM =====
- R6 [* -> 2.5.1]
- base64enc [* -> 0.1-3]
```

## 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"**

The screenshot shows the RStudio interface with the R console at the bottom. The console output is as follows:

```
R 3.4.4 · /cloud/project/

Restarting R session...

> renv::init()
This project already has a lockfile. What would you like to do?

1: Restore the project from the lockfile.
2: Discard the lockfile and re-initialize the project.
3: Activate the project without snapshotting or installing any packages.
4: Abort project initialization.

Selection: 1
```

The file explorer on the right shows the project structure:

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	

The source editor shows the following R Markdown code:

```
1 ---
2 title: "Hands on Exercise Reproducible Work"
3 author: "Xavier"
4 date: "2023-02-25"
5 output: html_document
6 ---
7
8 {r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10
11
12 # Session Reproducible Work
13
14 Monday Feb 27, 2023. IL3-UB.
15
16 Related to: <https://seeds4c.org/reproduciblework2023>
17
18 ## Hands on Exercise
19
20 {r}
21 if (!require(readr)) {install.packages("readr")}
22 if (!require(dplyr)) {install.packages("dplyr")}
23 if (!require(tidyr)) {install.packages("tidyr")}
24
```

## 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:** Contains R Markdown code for a reproducible work session. The code includes a title, author, date, output format, and a session setup block. A warning message at the top indicates that packages `dplyr` and `tidyr` are required but not installed.
- Environment Panel:** Shows the project files, including `.Rprofile`, `renv/`, and `renv.lock`.
- Files Panel:** Shows the project directory structure, including `.gitignore`, `.Rhistory`, `project.Rproj`, `README.md`, `recipes`, `renv.lock`, `ReproducibleWork_HandsOnExer...`, `renv`, and `.Rprofile`.
- Console:** Shows the output of the R session, including the installation of `tinytex`, `rmarkdown`, and `tidyr`. The console also shows the message "Restarting R session..." and "\* Project '/cloud/project' loaded. [renv 0.16.0]".

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

The screenshot displays the RStudio interface for a workspace named 'datascience2023'. The main editor shows an R Markdown document titled 'Hands on Exercise Reproducible Work' with the following content:

```
1 ---
2 title: "Hands on Exercise Reproducible Work"
3 author: "Xavier"
4 date: "2023-02-25"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 # Session Reproducible Work
13
14 Monday Feb 27, 2023. IL3-UB.
15
16 Related to: <https://seeds4c.org/reproduciblework2023>
17
18 ## Hands on Exercise
19
20 ```{r}
21 if (!require(readr)) {install.packages("readr")}
22 if (!require(dplyr)) {install.packages("dplyr")}
23 ```
```

The console shows the execution of the R code, including the installation of 'readr' and 'dplyr' packages, and the generation of the output file 'ReproducibleWork\_HandsOnExercise.R'. The output is:

```
R 3.4.4 > /cloud/project/
+ values_from = "VALOR_LLECTURA"
>
> data_wide
> # Save resulting dataset to disk
> write_csv(data_wide, "data_subset_d5_wide.csv")
> # Produce a simple R version of this R Markdown document
> knitr::purl("ReproducibleWork_HandsOnExercise.Rmd", documentation=2)

processing file: ReproducibleWork_HandsOnExercise.Rmd

output file: ReproducibleWork_HandsOnExercise.R

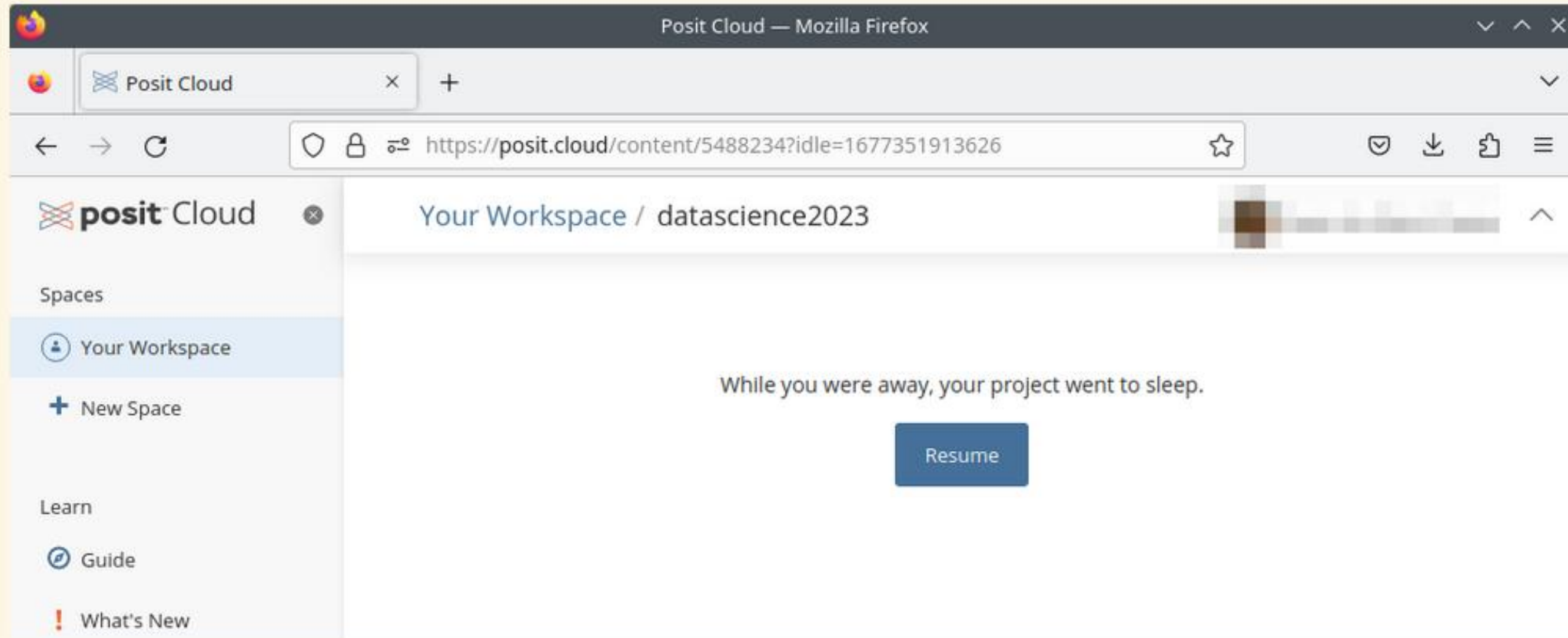
[1] "ReproducibleWork_HandsOnExercise.R"
>
```

The right-hand side of the interface shows the 'Environment' pane with a list of files and their sizes:

File	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
data_smc.csv.bz2	50.2 MB
data_smc.csv	613.3 MB
data_subset_all.csv	61.3 MB
data_subset_d5_wide.csv	48.6 KB
ReproducibleWork_HandsOnExer...	2.8 KB

# 6.6. Additional info

Project (Container) goes to sleep on inactivity



# Thanks

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

Slides available at:  
<https://seeds4c.org/reproduciblework2023>



Unless elsewhere noted, contents of this web site are released under a [Creative Commons](#)

[Commons](#) license.

The screenshot displays the Posit Cloud interface in a Mozilla Firefox browser window. The main workspace is titled "Your Workspace Project" and "datascience2023". The interface includes a sidebar with navigation options like "Spaces", "Learn", and "Help". The central area shows an R script editor with code for data manipulation using dplyr. Below the editor is a data viewer showing a tibble with columns "DATA\_LECTURA", "T", and "Pn". The console at the bottom shows terminal output for system information, including "aws Virtualization (Amazon Web Services)" and "Ubuntu 20.04.5 LTS Linux distro in container". The right sidebar shows the "Environment" pane with a list of R versions (4.2.2, 4.1.3, 4.0.5, 3.6.3, 3.5.3, 3.4.4) and the "Files" pane showing project files like ".gitignore", ".Rhistory", ".Rprofile", "project.Rproj", "README.md", "recipes", "renv", and "renv.lock". Red annotations with numbers 1-6 highlight specific features: 1 points to "aws Virtualization (Amazon Web Services)", 2 points to the R version list, 3 points to "renv" and "renv.lock", 4 points to the "Environment" pane header, 5 points to the R script code, and 6 points to the "Help" button in the Files pane.