practical{renv}

Shannon Pileggi

posit::conf(2025)

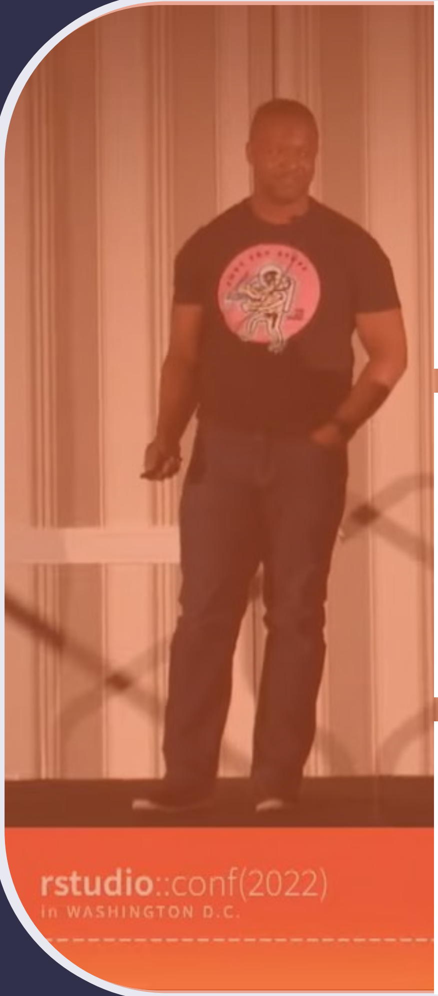
renv: Project Environments for R

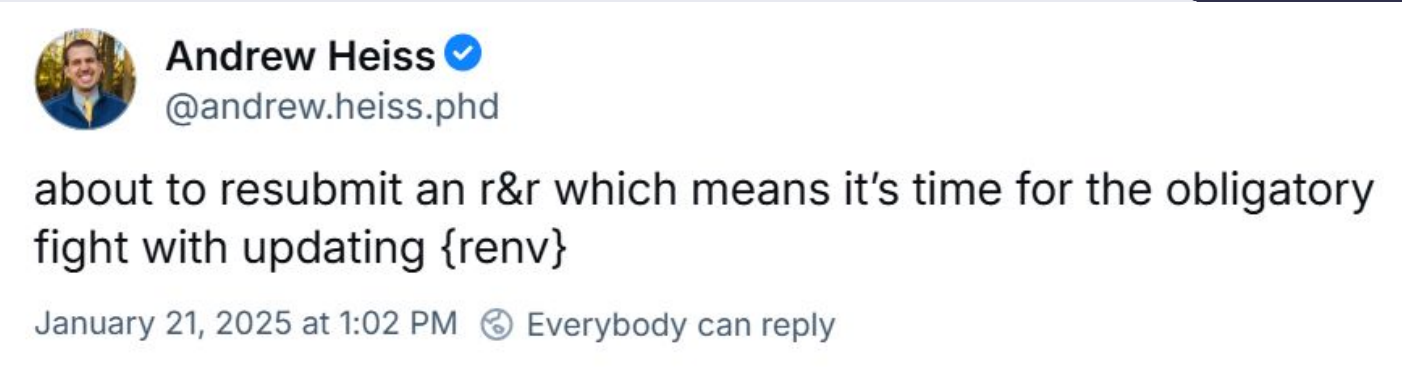
Kevin Ushey

2020-01-30

You should use renv.

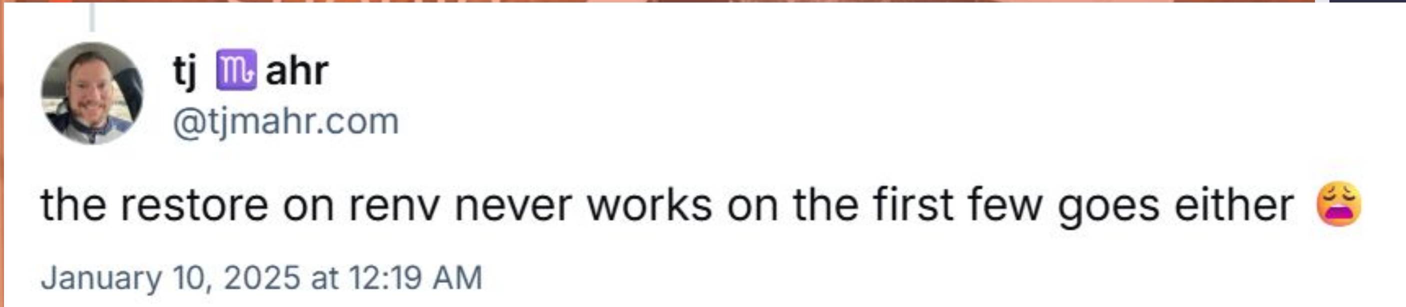
rstudio::conf(2022)
in WASHINGTON D.C.




Andrew Heiss ✓
@andrew.heiss.phd

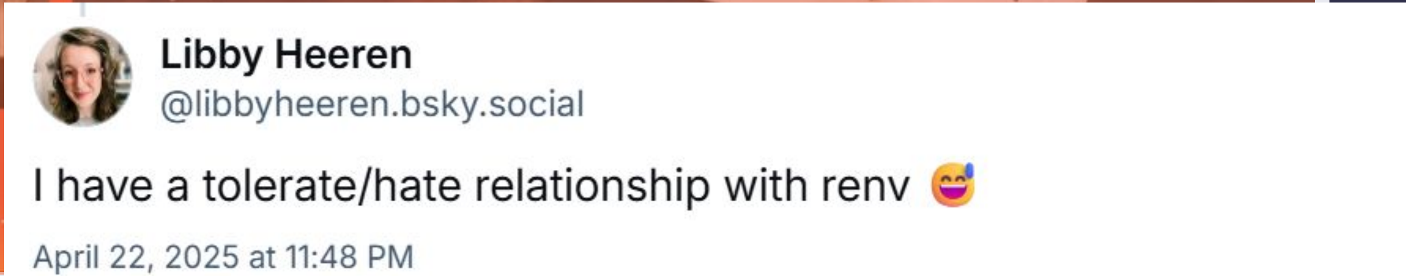
about to resubmit an r&r which means it's time for the obligatory fight with updating {renv}

January 21, 2025 at 1:02 PM · Everybody can reply


tj ♏ ahr
@tjmahr.com

the restore on renv never works on the first few goes either 😩

January 10, 2025 at 12:19 AM


Libby Heeren
@libbyheeren.bsky.social

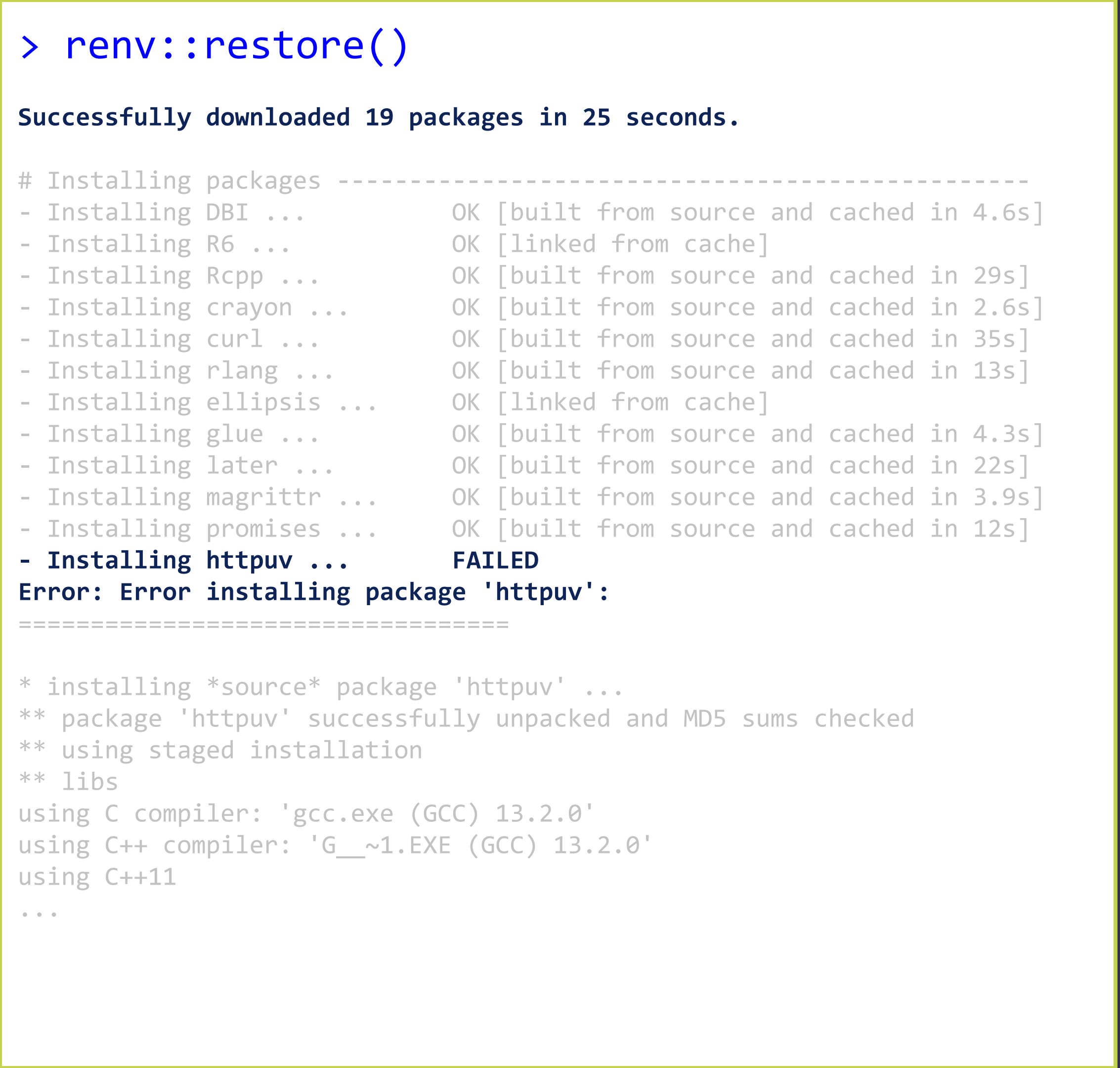
I have a tolerate/hate relationship with renv 😅

April 22, 2025 at 11:48 PM

Kevin Ushey
software engineer

E. David Aja
software engineer

Shannon Pileggi
data scientist



```
> renv::restore()

Successfully downloaded 19 packages in 25 seconds.

# Installing packages ------------------------------------------------
- Installing DBI ...            OK [built from source and cached in 4.6s]
- Installing R6 ...             OK [linked from cache]
- Installing Rcpp ...           OK [built from source and cached in 29s]
- Installing crayon ...         OK [built from source and cached in 2.6s]
- Installing curl ...           OK [built from source and cached in 35s]
- Installing rlang ...          OK [built from source and cached in 13s]
- Installing ellipsis ...       OK [linked from cache]
- Installing glue ...           OK [built from source and cached in 4.3s]
- Installing later ...          OK [built from source and cached in 22s]
- Installing magrittr ...       OK [built from source and cached in 3.9s]
- Installing promises ...       OK [built from source and cached in 12s]
- Installing httpuv ...         FAILED
Error: Error installing package 'httpuv':
==================================

* installing *source* package 'httpuv' ...
** package 'httpuv' successfully unpacked and MD5 sums checked
** using staged installation
** libs
using C compiler: 'gcc.exe (GCC) 13.2.0'
using C++ compiler: 'G__~1.EXE (GCC) 13.2.0'
using C++11
...
```

this talk
is for
frustrated
{renv} users
and
potential
{renv} users

```
> renv::restore()

Successfully downloaded 19 packages in 25 seconds.

# Installing packages ------------------------------------------------
- Installing DBI ...            OK [built from source and cached in 4.6s]
- Installing R6 ...             OK [linked from cache]
- Installing Rcpp ...           OK [built from source and cached in 29s]
- Installing crayon ...         OK [built from source and cached in 2.6s]
- Installing curl ...           OK [built from source and cached in 35s]
- Installing rlang ...          OK [built from source and cached in 13s]
- Installing ellipsis ...       OK [linked from cache]
- Installing glue ...           OK [built from source and cached in 4.3s]
- Installing later ...          OK [built from source and cached in 22s]
- Installing magrittr ...       OK [built from source and cached in 3.9s]
- Installing promises ...       OK [built from source and cached in 12s]
- Installing httpuv ...         FAILED
Error: Error installing package 'httpuv':
==================================

* installing *source* package 'httpuv' ...
** package 'httpuv' successfully unpacked and MD5 sums checked
** using staged installation
** libs
using C compiler: 'gcc.exe (GCC) 13.2.0'
using C++ compiler: 'G__~1.EXE (GCC) 13.2.0'
using C++11
...
```

```
> renv::restore()

Successfully downloaded 19 packages in 25 seconds.

# Installing packages ------------------------------------------------
- Installing DBI ...          OK [built from source and cached in 4.6s]
- Installing R6 ...           OK [linked from cache]
- Installing Rcpp ...         OK [built from source and cached in 29s]
- Installing crayon ...       OK [built from source and cached in 2.6s]
- Installing curl ...         OK [built from source and cached in 35s]
- Installing rlang ...        OK [built from source and cached in 13s]
- Installing ellipsis ...     OK [linked from cache]
- Installing glue ...         OK [built from source and cached in 4.3s]
- Installing later ...        OK [built from source and cached in 22s]
- Installing magrittr ...     OK [built from source and cached in 3.9s]
- Installing promises ...     OK [built from source and cached in 12s]
- Installing httpuv ...       FAILED
Error: Error installing package 'httpuv':
==================================

* installing *source* package 'httpuv' ...
** package 'httpuv' successfully unpacked and MD5 sums checked
** using staged installation
** libs
using C compiler: 'gcc.exe (GCC) 13.2.0'
using C++ compiler: 'G__~1.EXE (GCC) 13.2.0'
using C++11
...
```

```
> renv::restore()

Successfully downloaded 19 packages in 25 seconds.

# Installing packages ----------------------------------------------------
- Installing DBI ...           OK [built from source and cached in 4.6s]
- Installing R6 ...            OK [linked from cache]
- Installing Rcpp ...          OK [built from source and cached in 29s]
- Installing crayon ...        OK [built from source and cached in 2.6s]
- Installing curl ...          OK [built from source and cached in 35s]
- Installing rlang ...         OK [built from source and cached in 13s]
- Installing ellipsis ...      OK [linked from cache]
- Installing glue ...          OK [built from source and cached in 4.3s]
- Installing later ...         OK [built from source and cached in 22s]
- Installing magrittr ...      OK [built from source and cached in 3.9s]
- Installing promises ...      OK [built from source and cached in 12s]
- Installing httpuv ...        FAILED
Error: Error installing package 'httpuv':
==================================

* installing *source* package 'httpuv' ...
** package 'httpuv' successfully unpacked and MD5 sums checked
** using staged installation
** libs
using C compiler: 'gcc.exe (GCC) 13.2.0'
using C++ compiler: 'G__~1.EXE (GCC) 13.2.0'
using C++11
...
```

Evaluate project state against current R configuration
renv::restore()
OPENING A {RENV}-BACKED PROJECT
JAKE-CLARK.TUMBLR
imgflip.com

PENGU

how to make {renv} actually work

~~how to make {renv} actually work~~

practical {renv}

goal

empower {renv} users to successfully restore a project environment, by either diagnosing or avoiding restore errors

intro to {renv}

package installation

projects over time

intro to
{renv}

package
installation

projects
over time

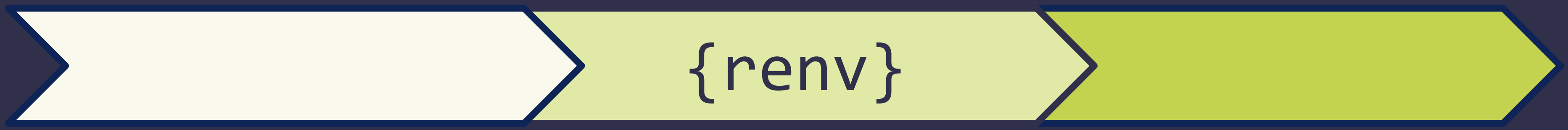
The renv package helps you create reproducible environments for your R projects.

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

The renv package helps you create reproducible environments for your R projects.

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

reproducibility spectrum



code

code
pkg version

code
pkg version
R version
sys. dep.
OS

The renv package helps you create reproducible environments for your R projects.

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

jsonlite-example initialized 2025-02-01

R version 4.4.2 (2024-10-31) -- "Pile of Leaves"

script.R

```r
library(jsonlite)
```

```
>
```

✗ {renv}

```
.
├── jsonlite-example.Rproj
├── script.R
```

jsonlite-example initialized 2025-02-01

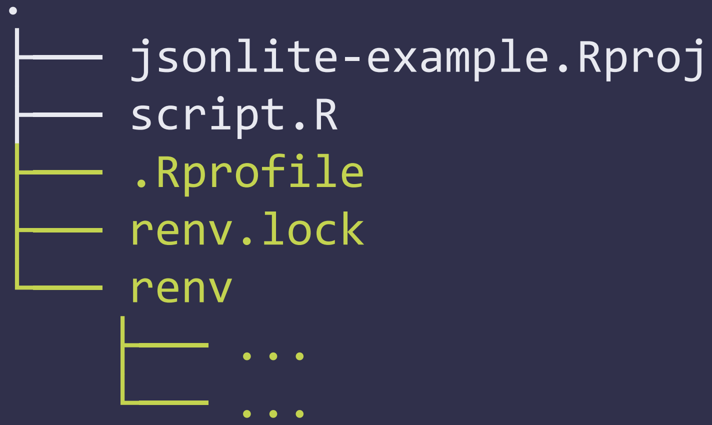
R version 4.4.2 (2024-10-31) -- "Pile of Leaves"

script.R

```
library(jsonlite)
```

```
> renv::init()

The following package(s) will be updated in
the lockfile:

# CRAN --------------------------------
- jsonlite    [* -> 1.8.9]
- renv        [* -> 1.1.0]

The version of R recorded in the lockfile
will be updated:
- R           [* -> 4.4.2]
```

✓{renv}

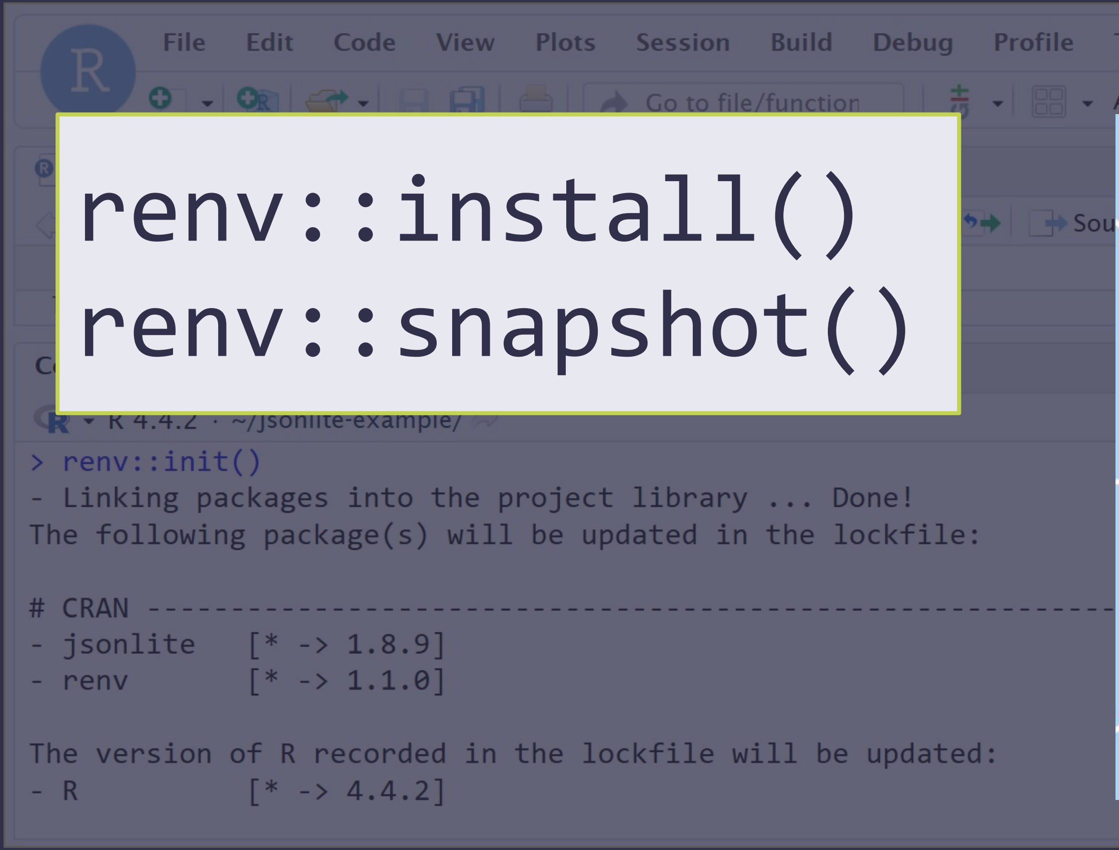


```
.
├── jsonlite-example.Rproj
├── script.R
├── .Rprofile
├── renv.lock
└── renv
    ├── ...
    └── ...
```

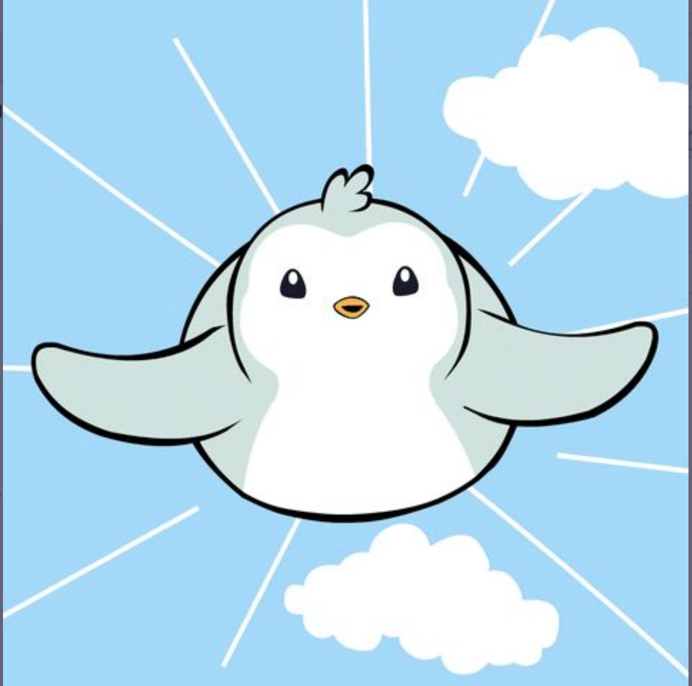
jsonlite-example

{renv}

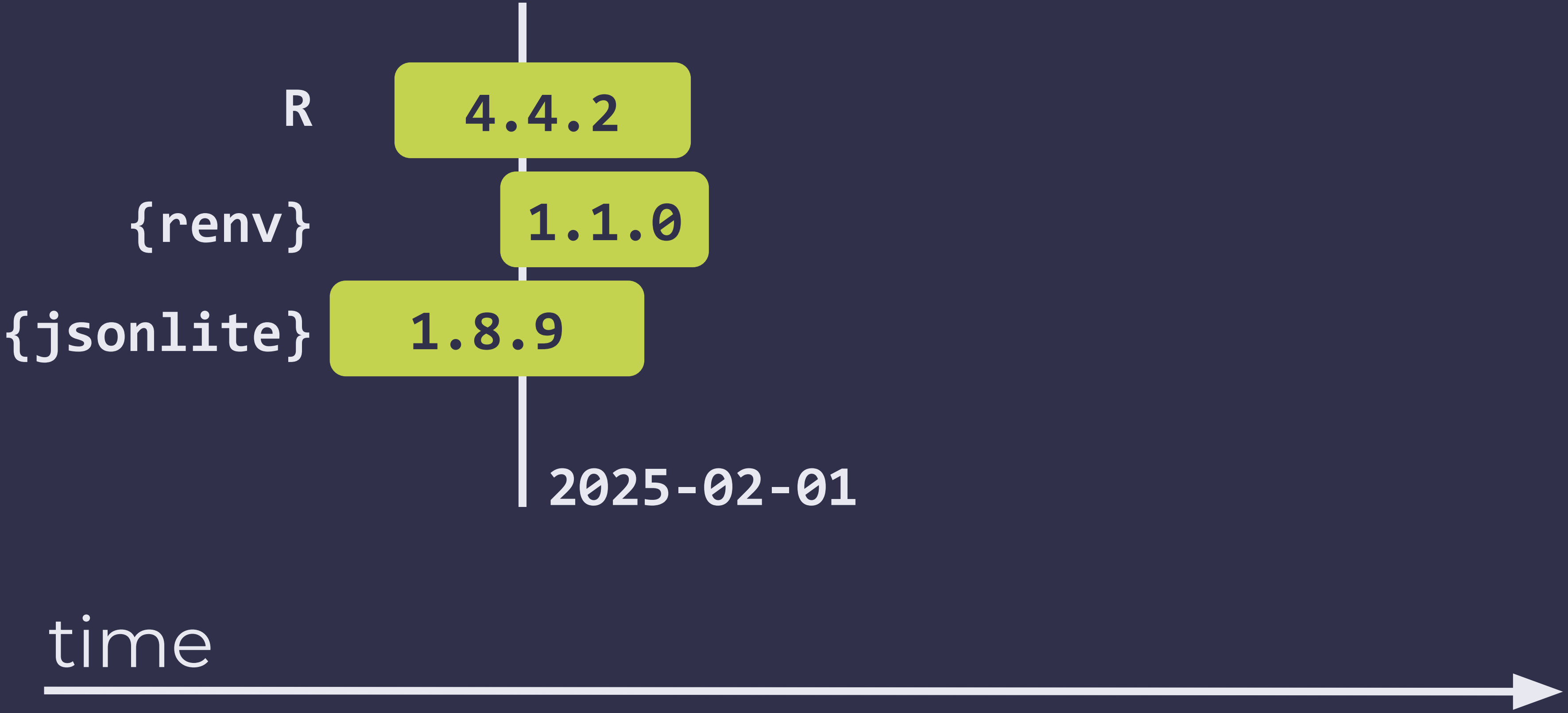
```
renv::install()
renv::snapshot()
```



```
> renv::init()
- Linking packages into the project library ... Done!
The following package(s) will be updated in the lockfile:

# CRAN --------------------------------------------------
- jsonlite   [* -> 1.8.9]
- renv       [* -> 1.1.0]

The version of R recorded in the lockfile will be updated:
- R          [* -> 4.4.2]
```



initial project



intro to {renv}

package installation

projects over time

library: a directory containing installed packages

jsonlite-example

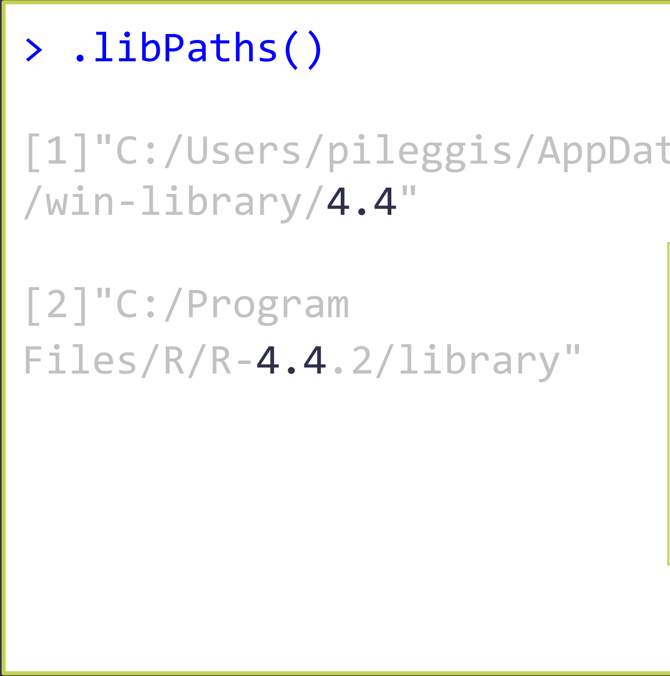
✕{renv}

```
> .libPaths()

[1]"C:/Users/pileggis/AppData/Local/R
/win-library/4.4"

[2]"C:/Program
Files/R/R-4.4.2/library"
```

library: a directory containing installed packages

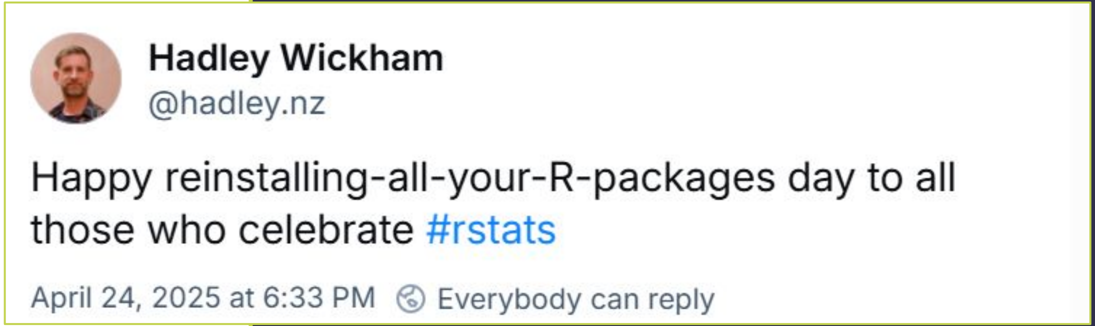
jsonlite-example

✖ {renv}



```
> .libPaths()

[1]"C:/Users/pileggis/AppData/Local/R
/win-library/4.4"

[2]"C:/Program
Files/R/R-4.4.2/library"
```

R 4.5.0 released



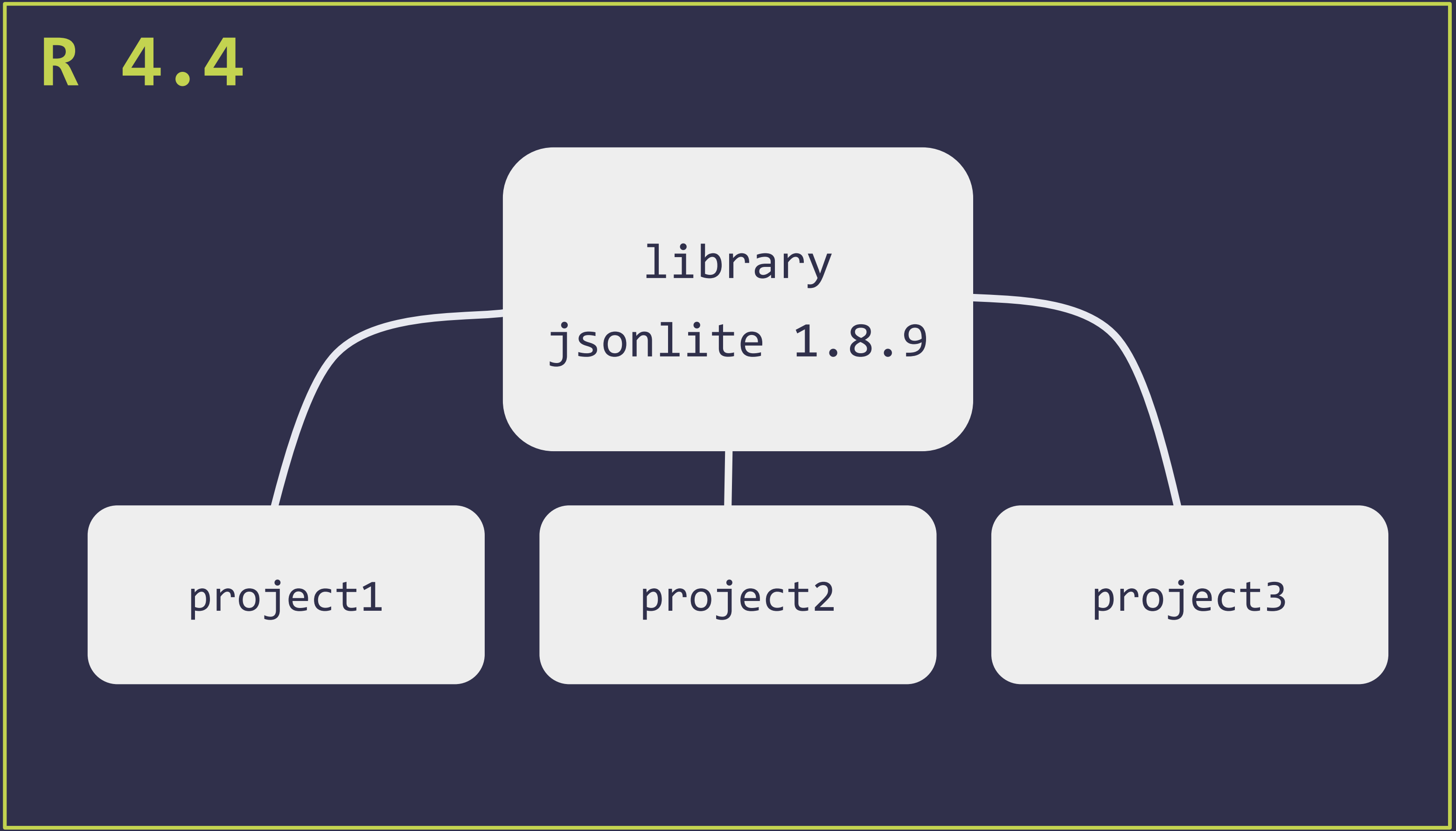
Hadley Wickham
@hadley.nz

Happy reinstalling-all-your-R-packages day to all those who celebrate #rstats

April 24, 2025 at 6:33 PM · Everybody can reply

shared package environment

✕{renv}



library: a directory containing installed packages

jsonlite-example

✗ {renv}

```
> .libPaths()

[1]"C:/Users/pileggis/AppData/Local/R
/win-library/4.4"

[2]"C:/Program
Files/R/R-4.4.2/library"
```

✓ {renv}

```
> .libPaths()

[1]"C:/Users/pileggis/Documents/gh-pe
rsonal/jsonlite-example/renv/
library/windows/R-4.4/x86_64-w64-ming
w32"

[2]"C:/Users/pileggis/AppData/Local/R
/cache/R/renv/sandbox/windows/
R-4.4/x86_64-w64-mingw32/6698a5f3"
```

library: a directory containing installed packages

jsonlite-example

✗{renv}

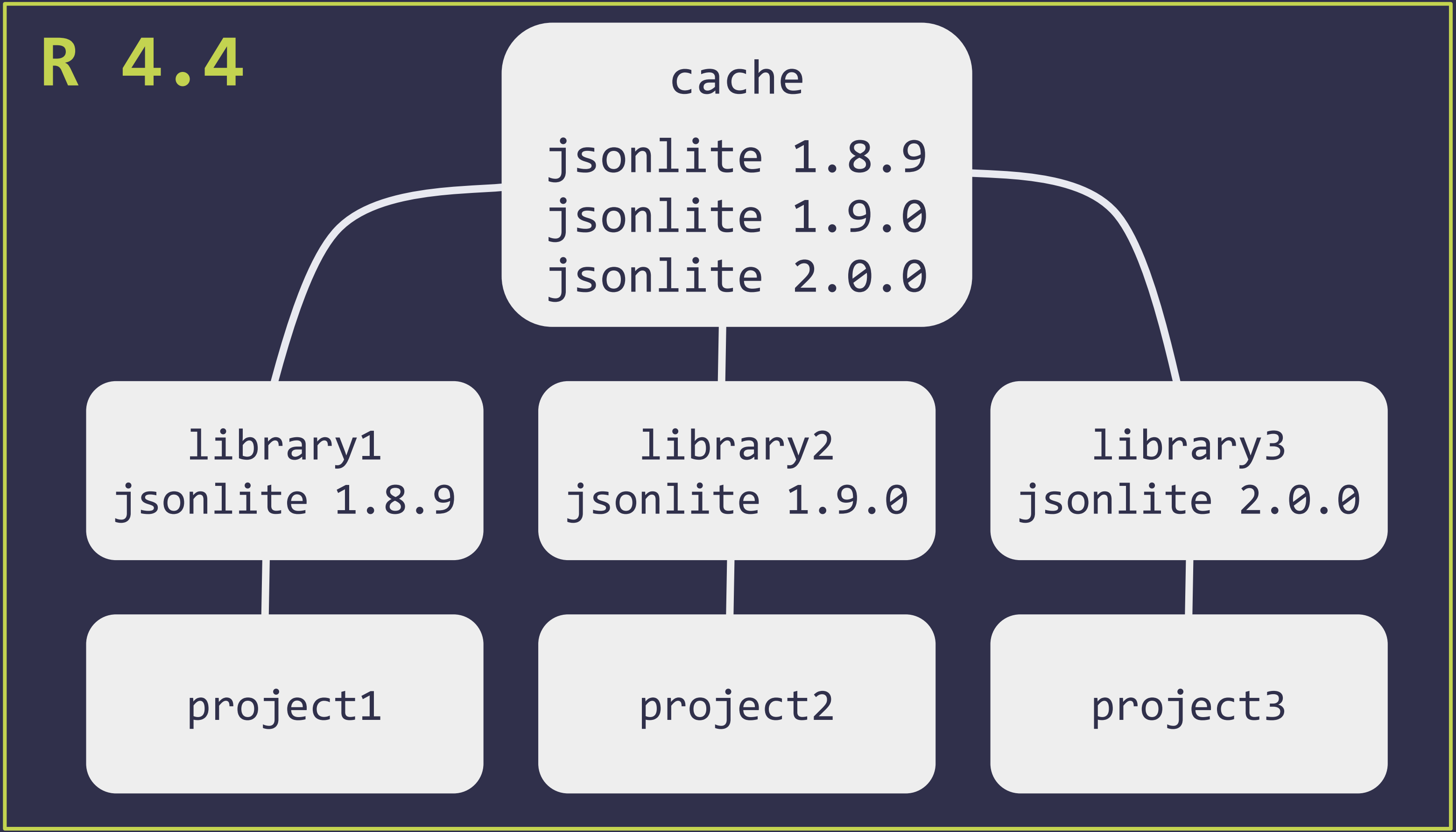
```
> .libPaths()

[1]"C:/Users/pileggis/AppData/Local/R
/win-library/4.4"

[2]"C:/Program
Files/R/R-4.4.2/library"
```

✓{renv}

```
> .libPaths()

[1]"C:/Users/pileggis/Documents/gh-pe
rsonal/jsonlite-example/renv/
library/windows/R-4.4/x86_64-w64-ming
w32"

[2]"C:/Users/pileggis/AppData/Local/R
/cache/R/renv/sandbox/windows/
R-4.4/x86_64-w64-mingw32/6698a5f3"
```

isolated package environment

✓ {renv}



R 4.4

cache

jsonlite 1.8.9
jsonlite 1.9.0
jsonlite 2.0.0

library1
jsonlite 1.8.9

library2
jsonlite 1.9.0

library3
jsonlite 2.0.0

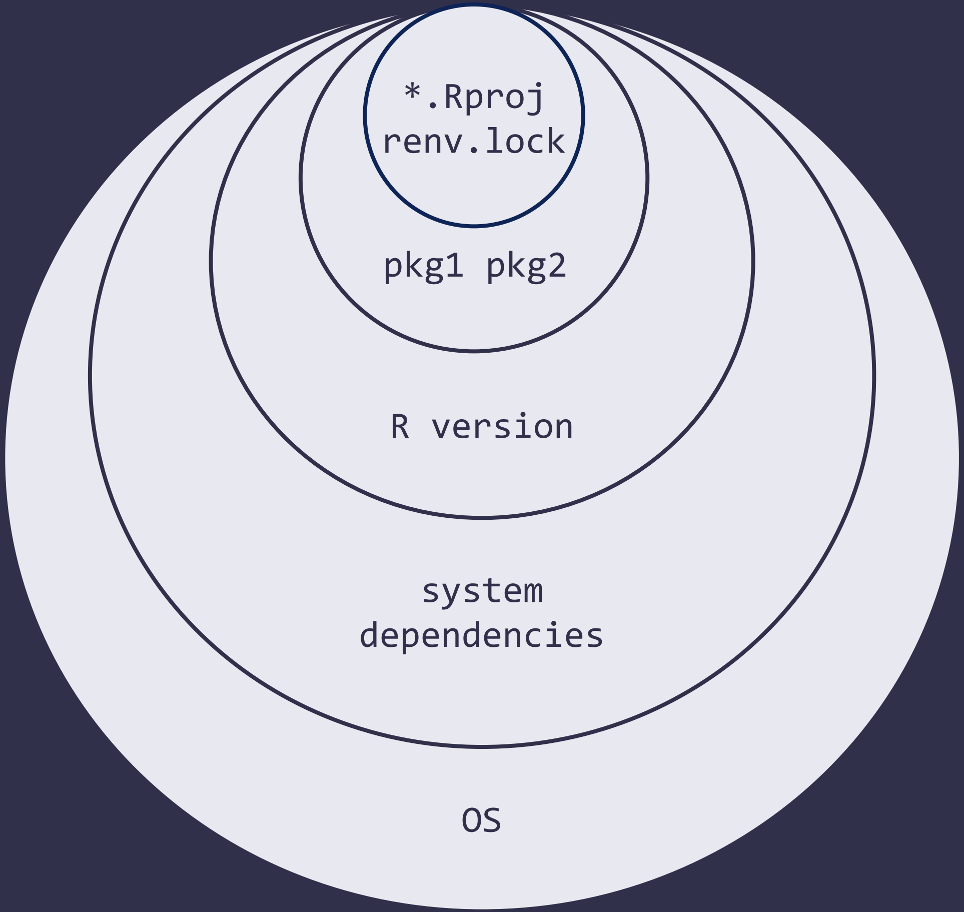
project1

project2

project3

project layers

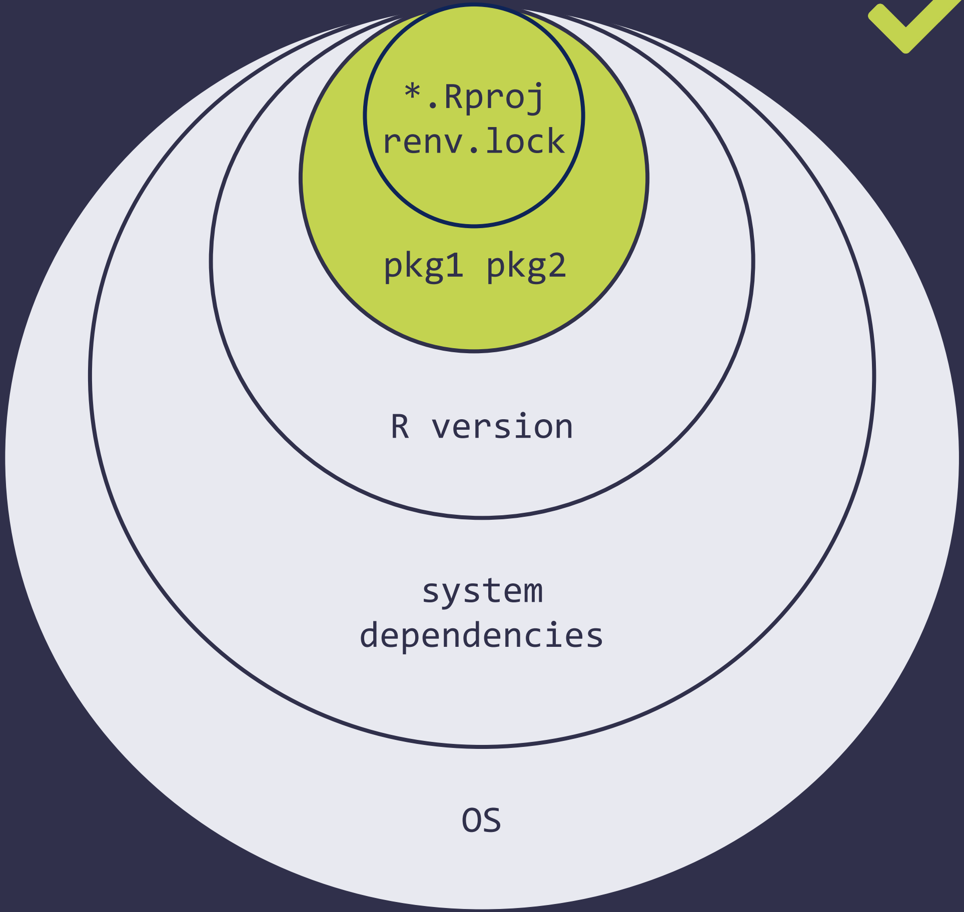
project1



project layers

✓ {renv}

project1



The renv.lock is a json file containing information from the installed package's `DESCRIPTION` file.

```
{
  "R": {
    "Version": "4.4.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cloud.r-project.org"
      }
    ]
  },
  "Packages": {
    "jsonlite": {
      "Package": "jsonlite",
      "Version": "1.8.9",
      "Source": "Repository",
      "NeedsCompilation": "yes",
      "Repository": "CRAN"
    },
    "renv": {
      "Package": "renv",
      "Version": "1.1.0",
      "Source": "Repository",
      "NeedsCompilation": "no",
      "Repository": "CRAN"
    }
  }
}
```

renv.lock

Version

```
{
  "R": {
    "Version": "4.4.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cloud.r-project.org"
      }
    ]
  },
  "Packages": {
    "jsonlite": {
      "Package": "jsonlite",
      "Version": "1.8.9",
      "Source": "Repository",
      "NeedsCompilation": "yes",
      "Repository": "CRAN"
    },
    "renv": {
      "Package": "renv",
      "Version": "1.1.0",
      "Source": "Repository",
      "NeedsCompilation": "no",
      "Repository": "CRAN"
    }
  }
}
```

renv.lock

Repositories

```
{
  "R": {
    "Version": "4.4.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cloud.r-project.org"
      }
    ]
  },
  "Packages": {
    "jsonlite": {
      "Package": "jsonlite",
      "Version": "1.8.9",
      "Source": "Repository",
      "NeedsCompilation": "yes",
      "Repository": "CRAN"
    },
    "renv": {
      "Package": "renv",
      "Version": "1.1.0",
      "Source": "Repository",
      "NeedsCompilation": "no",
      "Repository": "CRAN"
    }
  }
}
```

renv.lock

Repositories

CRAN
Posit public package manager (P3M)
Bioconductor
R-universe

```json
{
  "R": {
    "Version": "4.4.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cloud.r-project.org"
      }
    ]
  },
  "Packages": {
    "jsonlite": {
      "Package": "jsonlite",
      "Version": "1.8.9",
      "Source": "Repository",
      "NeedsCompilation": "yes",
      "Repository": "CRAN"
    },
    "renv": {
      "Package": "renv",
      "Version": "1.1.0",
      "Source": "Repository",
      "NeedsCompilation": "no",
      "Repository": "CRAN"
    }
  }
}
```

renv.lock

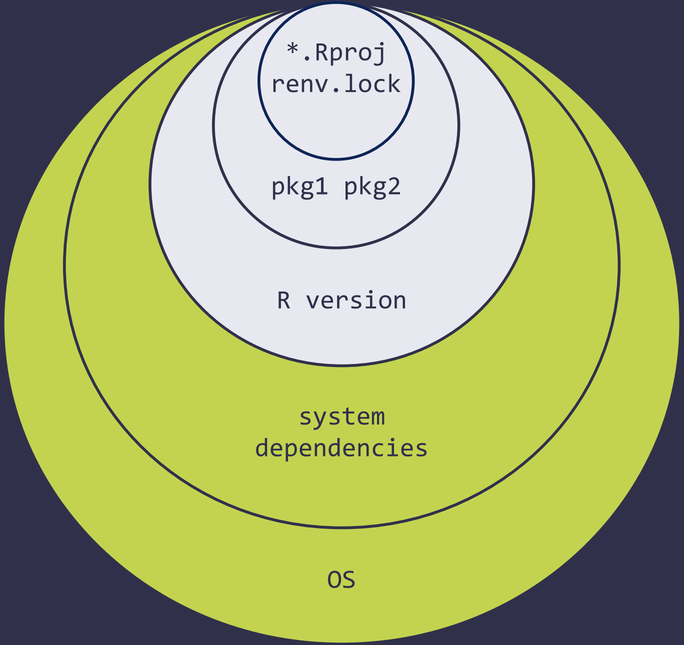
NeedsCompilation

```
{
  "R": {
    "Version": "4.4.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cloud.r-project.org"
      }
    ]
  },
  "Packages": {
    "jsonlite": {
      "Package": "jsonlite",
      "Version": "1.8.9",
      "Source": "Repository",
      "NeedsCompilation": "yes",
      "Repository": "CRAN"
    },
    "renv": {
      "Package": "renv",
      "Version": "1.1.0",
      "Source": "Repository",
      "NeedsCompilation": "no",
      "Repository": "CRAN"
    }
  }
}
```

renv.lock

NeedsCompilation

requires compiling tools outside of R



```
{
  "R": {
    "Version": "4.4.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cloud.r-project.org"
      }
    ]
  },
  "Packages": {
    "jsonlite": {
      "Package": "jsonlite",
      "Version": "1.8.9",
      "Source": "Repository",
      "NeedsCompilation": "yes",
      "Repository": "CRAN"
    },
    "renv": {
      "Package": "renv",
      "Version": "1.1.0",
      "Source": "Repository",
      "NeedsCompilation": "no",
      "Repository": "CRAN"
    }
  }
}
```

binaries: compiled R packages

specific to operating system, R version,
package version, and architecture

binaries: compiled R packages

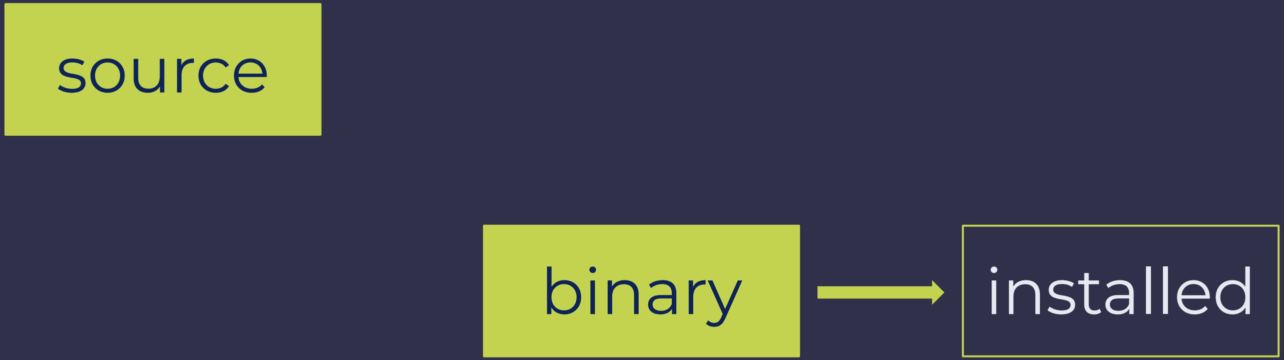
specific to operating system, R version,
package version, and architecture

source

binary

binaries: compiled R packages

specific to operating system, R version,
package version, and architecture



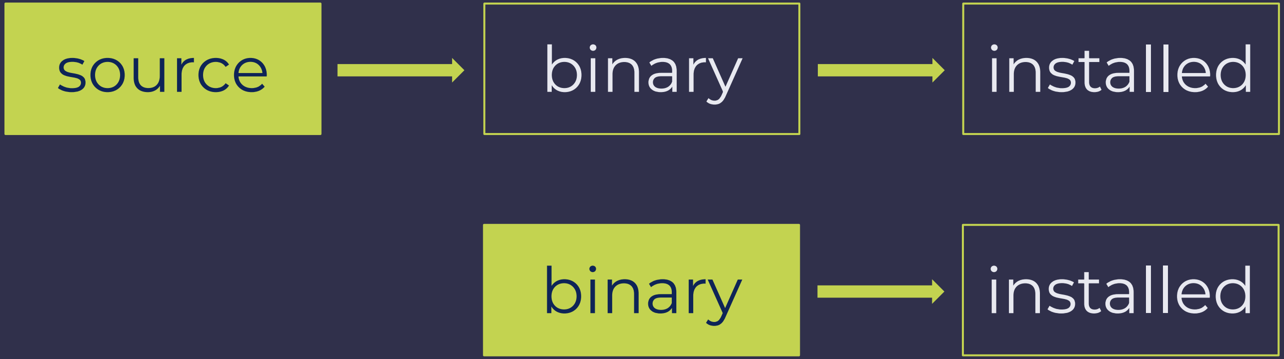
source

binary

installed

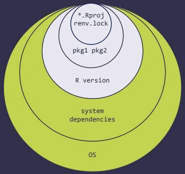
binaries: compiled R packages

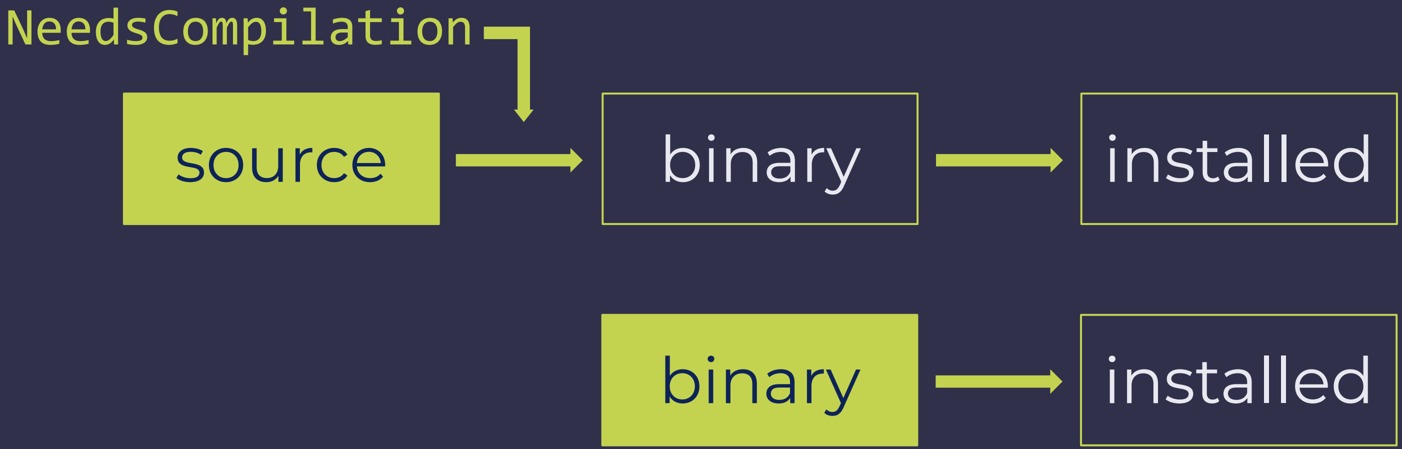
specific to operating system, R version,
package version, and architecture



binaries: compiled R packages

specific to operating system, R version,
package version, and architecture





NeedsCompilation

source

binary

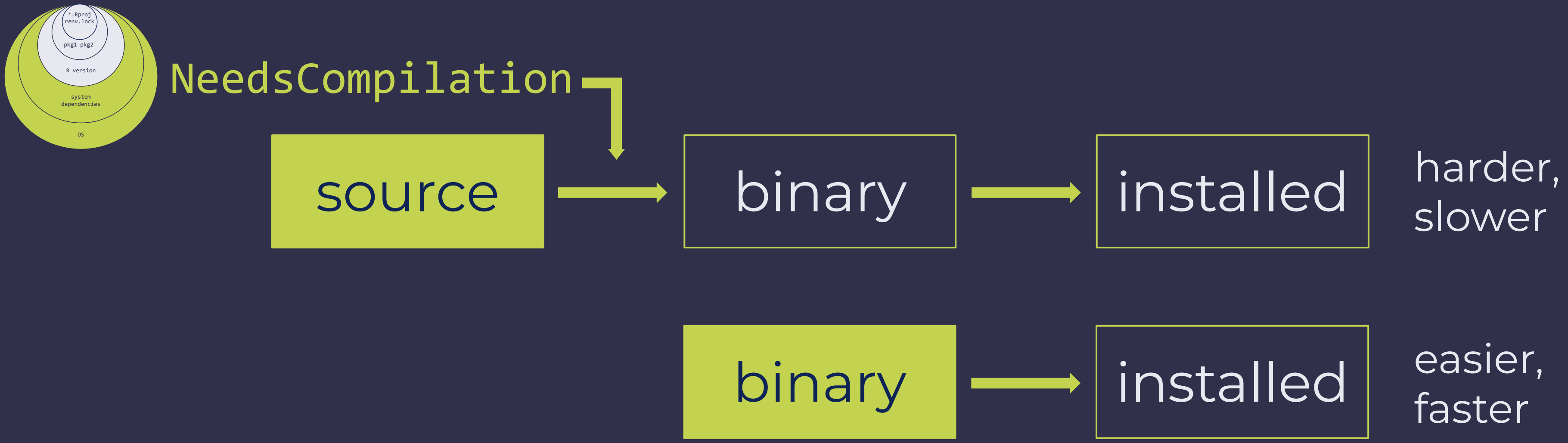
installed

binary

installed

binaries: compiled R packages

specific to operating system, R version, package version, and architecture



NeedsCompilation

source → binary → installed

harder, slower

binary → installed

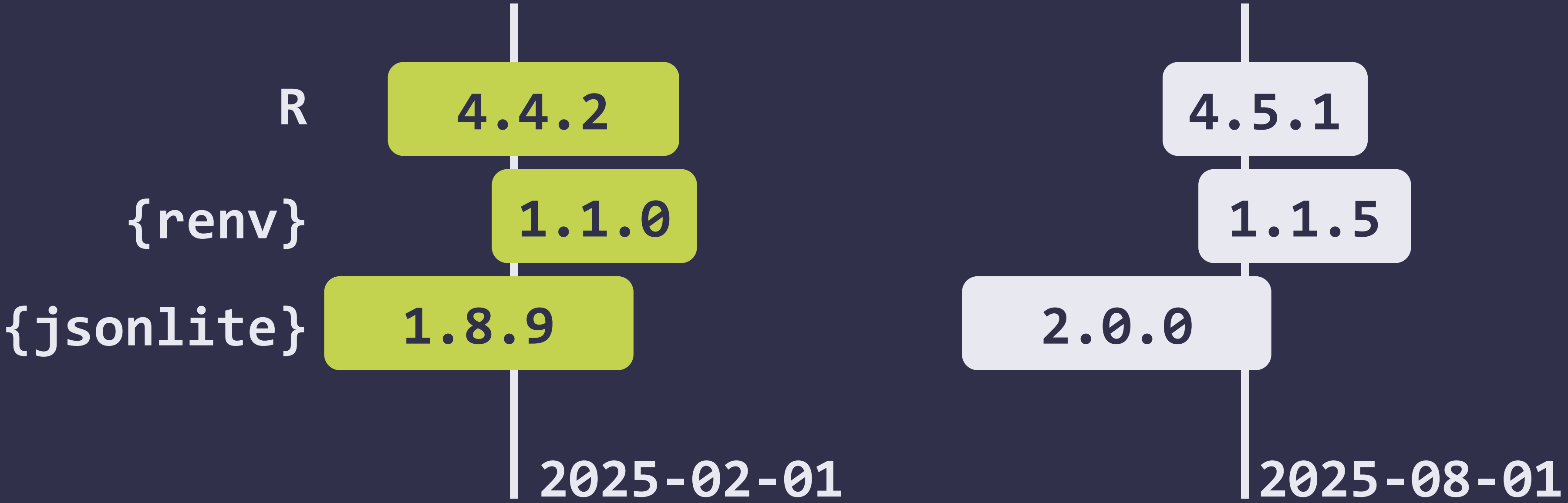
easier, faster

intro to
{renv}

package
installation

projects
over time

resume an active project



R 4.4.2 4.5.1

{renv} 1.1.0 1.1.5

{jsonlite} 1.8.9 2.0.0

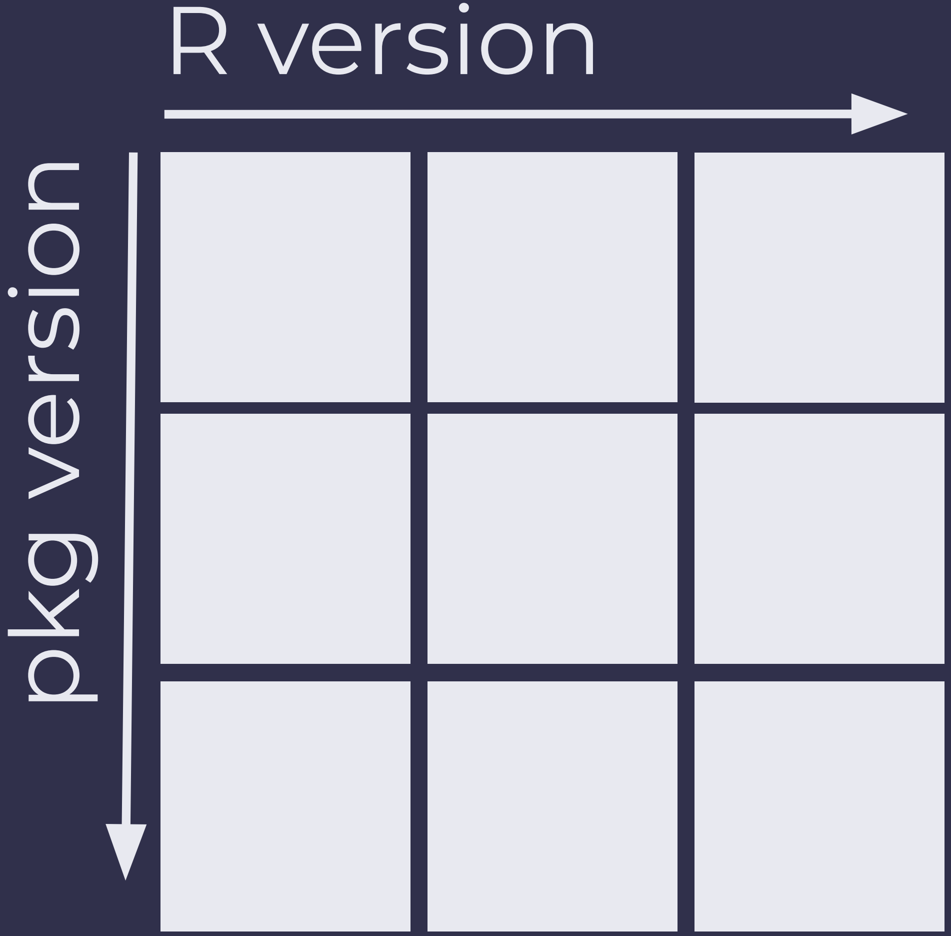
2025-02-01

2025-08-01

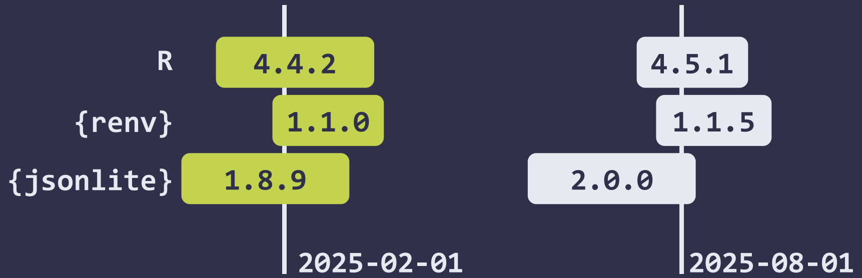
project decisions

1. freeze
2. manage
3. update

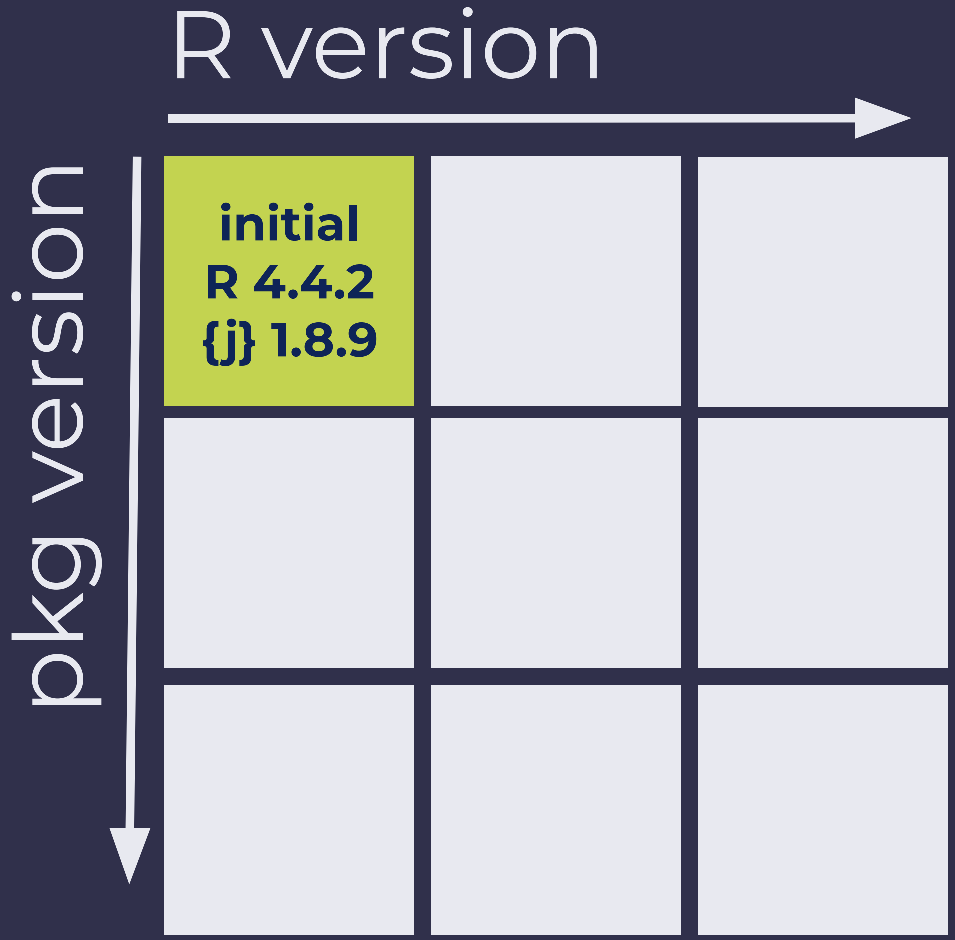
versions over time



versions over time



	2025-02-01	2025-08-01
R	4.4.2	4.5.1
{renv}	1.1.0	1.1.5
{jsonlite}	1.8.9	2.0.0



project decisions

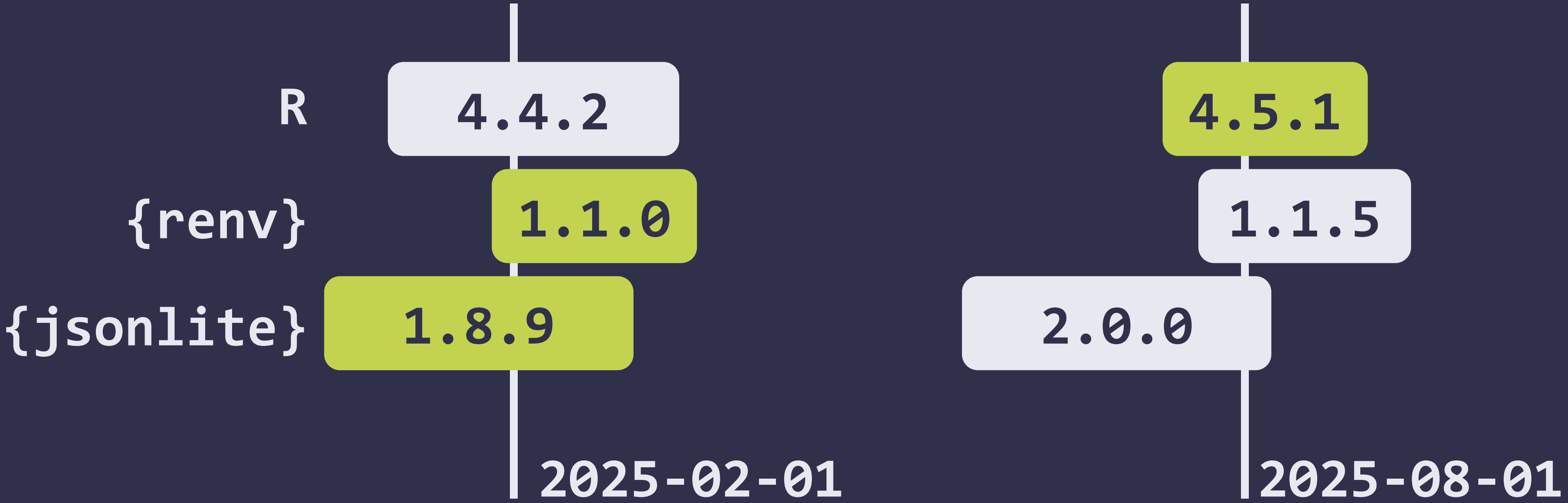
1 freeze

2 manage

3 update

resume project

can i...?



R 4.4.2 4.5.1

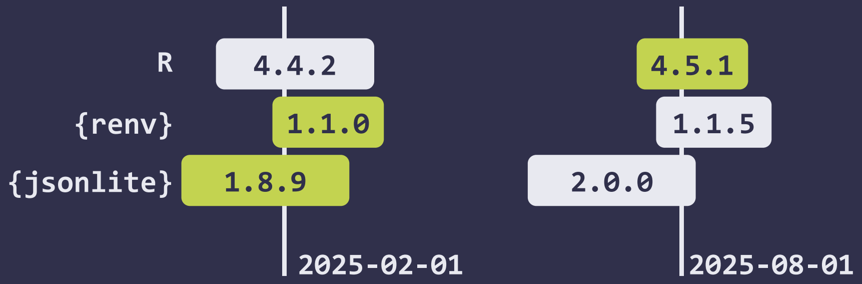
{renv} 1.1.0 1.1.5

{jsonlite} 1.8.9 2.0.0

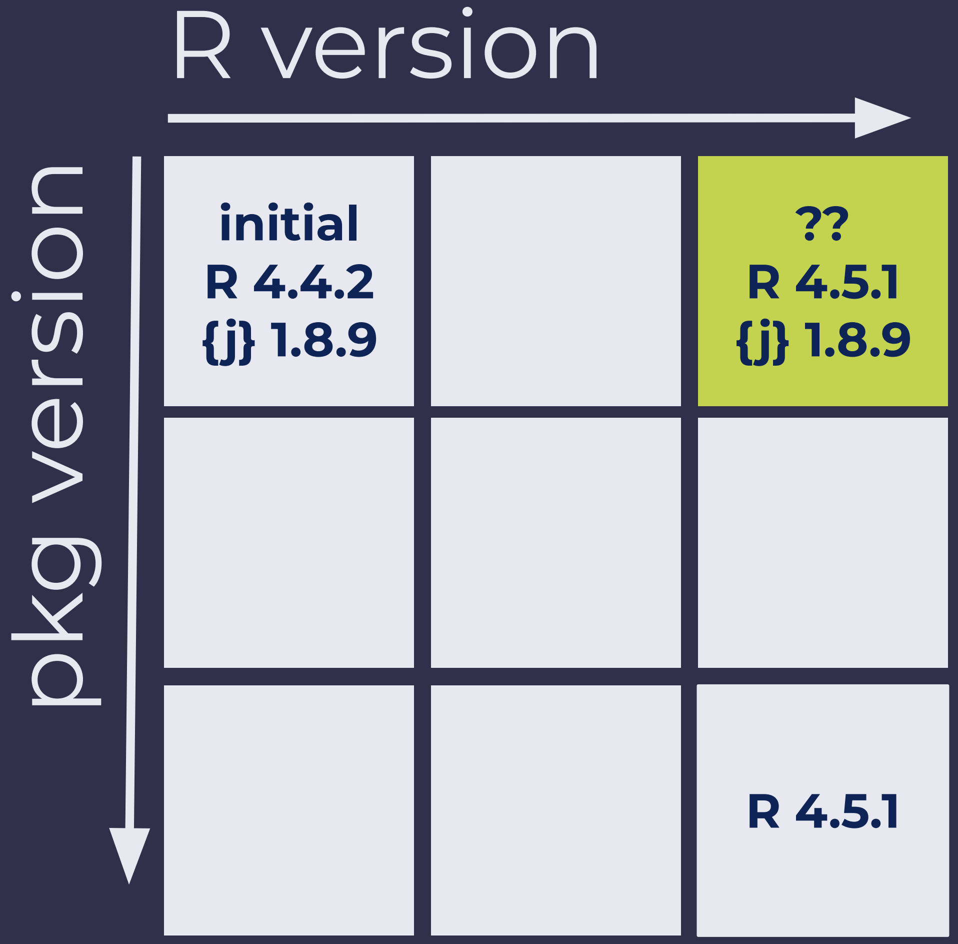
2025-02-01

2025-08-01

versions over time



	2025-02-01	2025-08-01
R	4.4.2	4.5.1
{renv}	1.1.0	1.1.5
{jsonlite}	1.8.9	2.0.0

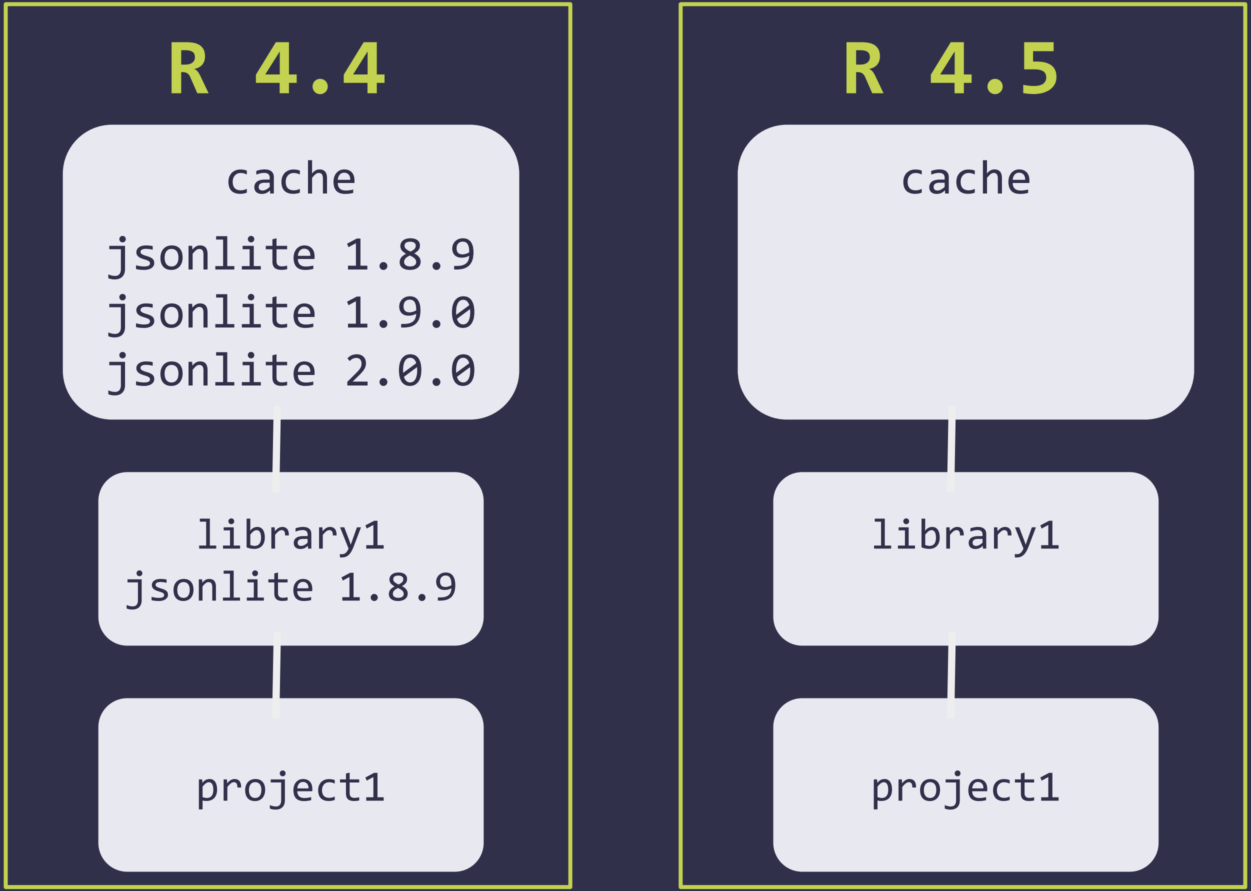


R version

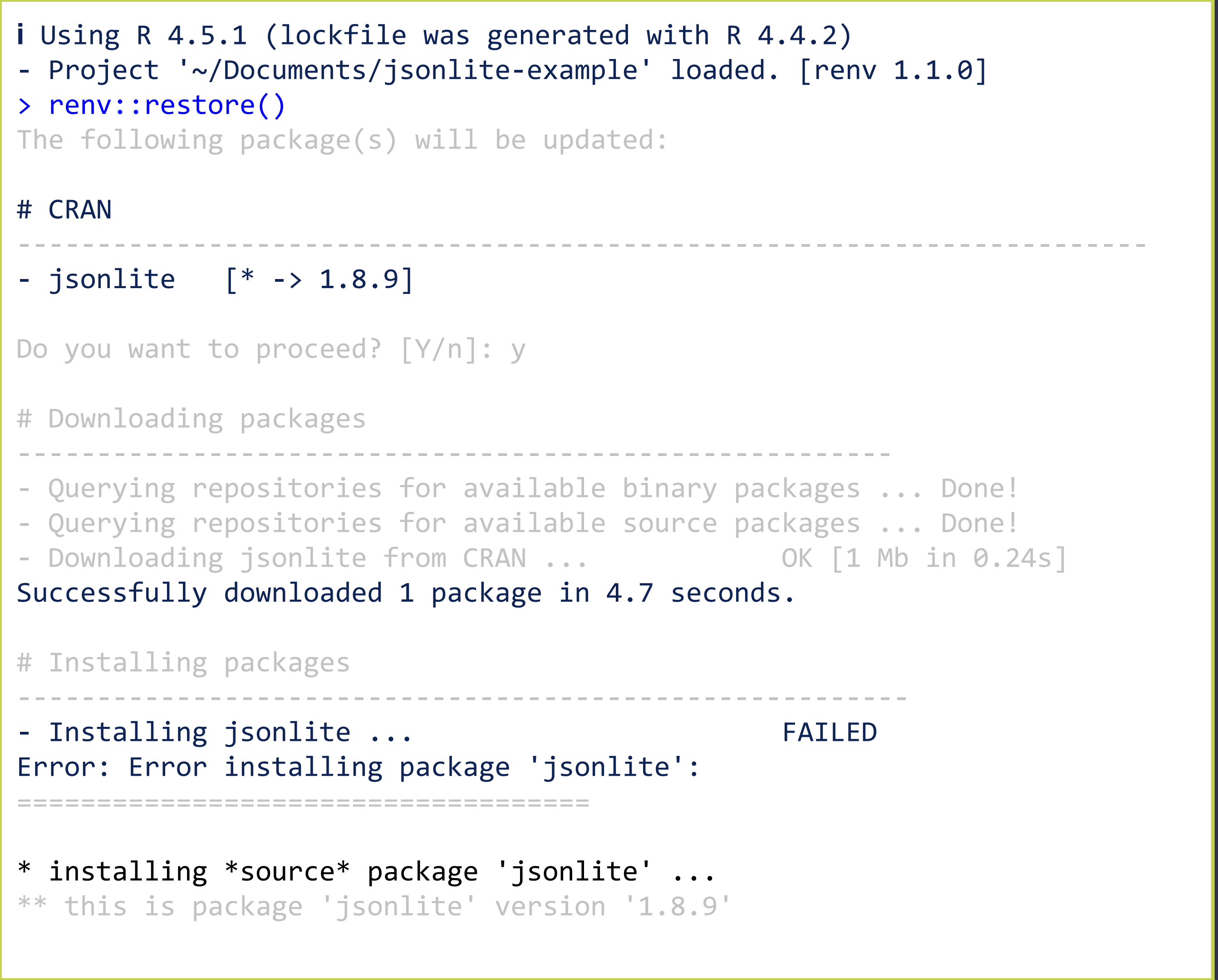
pkg version

initial R 4.4.2 {j} 1.8.9		?? R 4.5.1 {j} 1.8.9
		R 4.5.1

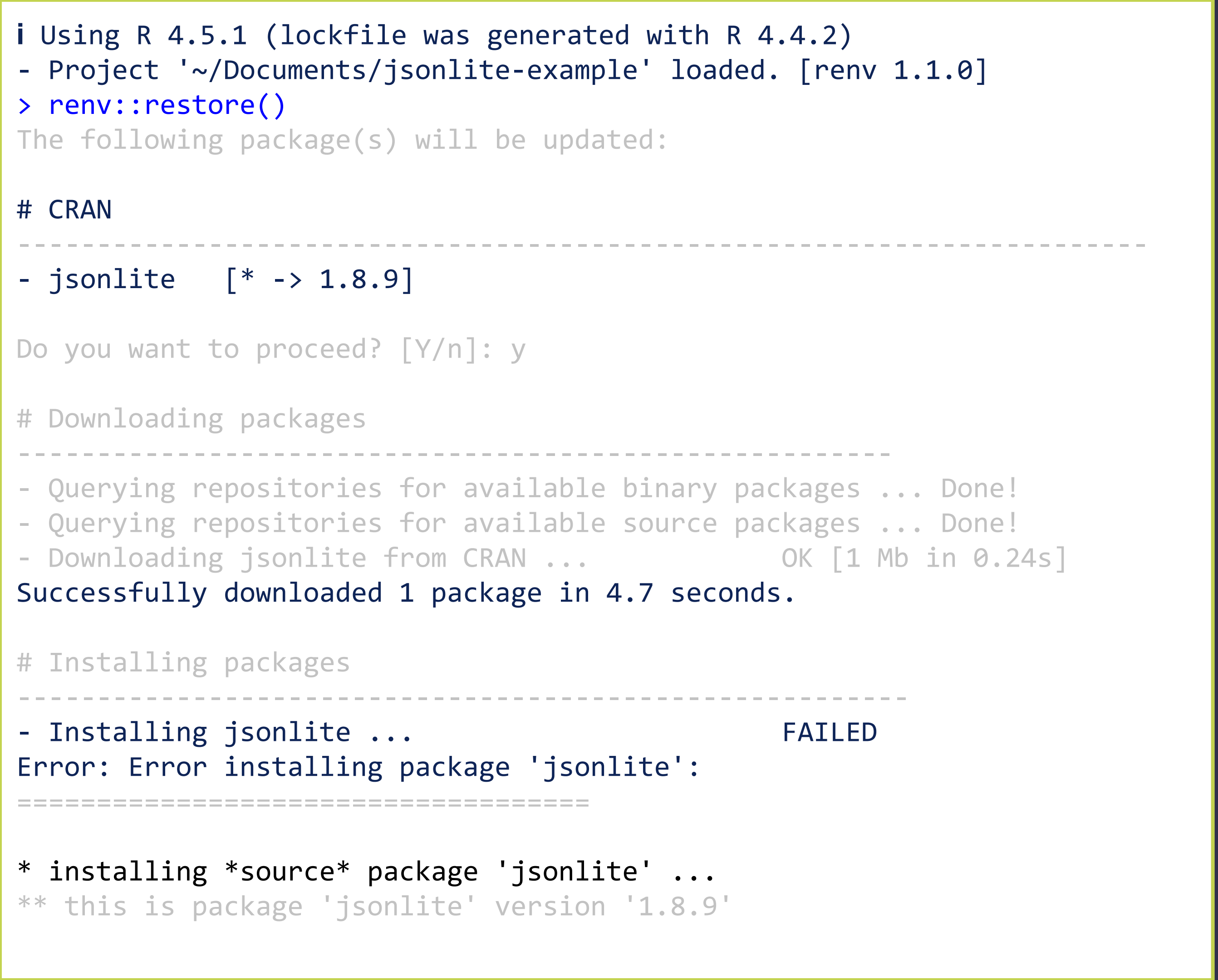
new R version, new cache



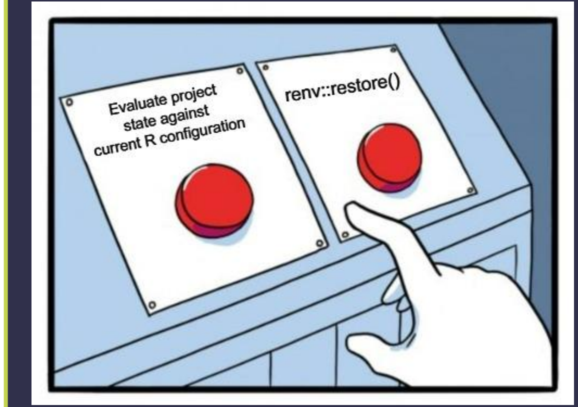
```
i Using R 4.5.1 (lockfile was generated with R 4.4.2)
- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
```

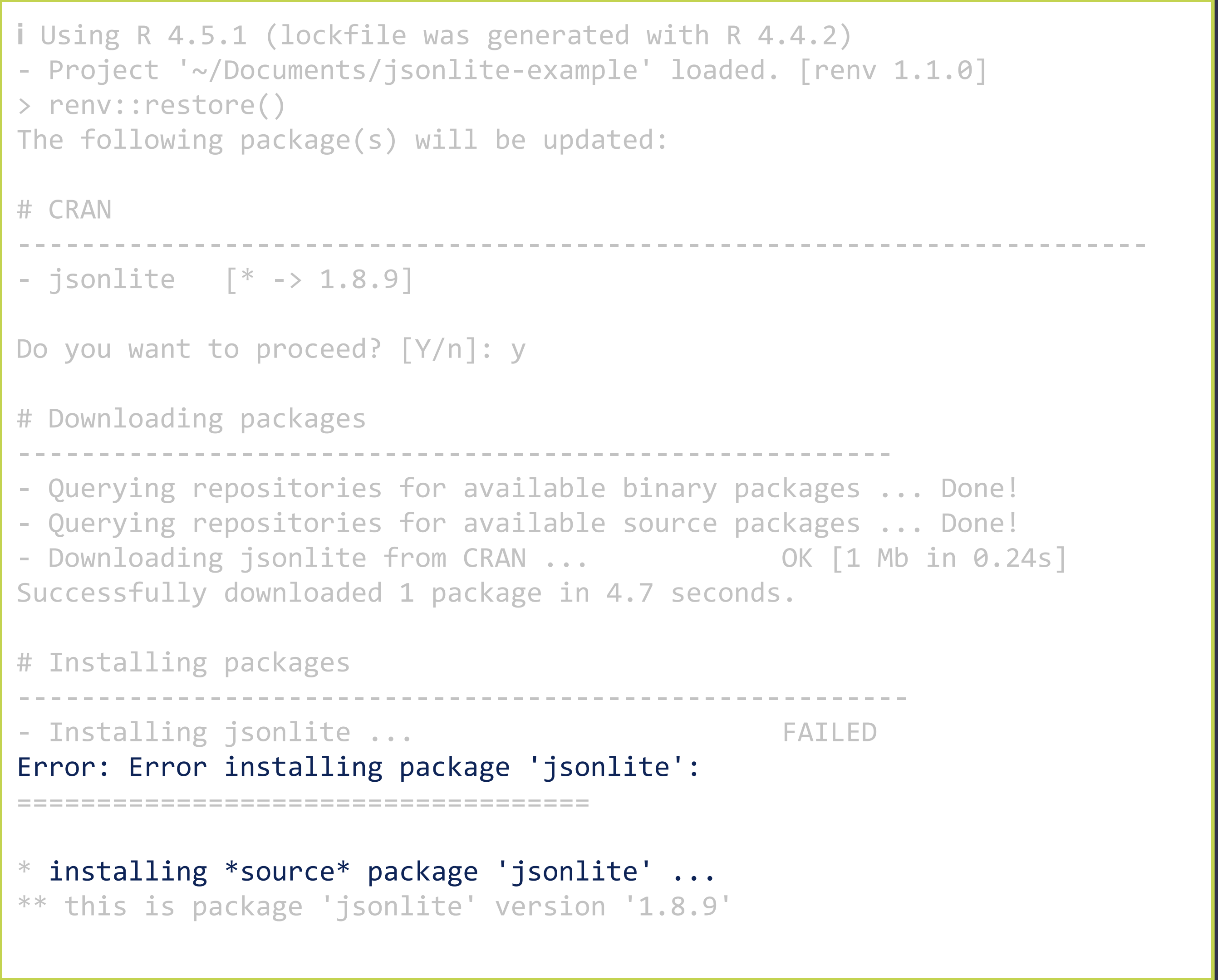


```
i Using R 4.5.1 (lockfile was generated with R 4.4.2)
- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
The following package(s) will be updated:

# CRAN
------------------------------------------------------------------------
- jsonlite   [* -> 1.8.9]

Do you want to proceed? [Y/n]: y

# Downloading packages
-------------------------------------------------------
- Querying repositories for available binary packages ... Done!
- Querying repositories for available source packages ... Done!
- Downloading jsonlite from CRAN ...            OK [1 Mb in 0.24s]
Successfully downloaded 1 package in 4.7 seconds.

# Installing packages
--------------------------------------------------------
- Installing jsonlite ...                       FAILED
Error: Error installing package 'jsonlite':
====================================

* installing *source* package 'jsonlite' ...
** this is package 'jsonlite' version '1.8.9'
```



```
i Using R 4.5.1 (lockfile was generated with R 4.4.2)
- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
The following package(s) will be updated:

# CRAN
------------------------------------------------------------------------
- jsonlite    [* -> 1.8.9]

Do you want to proceed? [Y/n]: y

# Downloading packages
--------------------------------------------------------
- Querying repositories for available binary packages ... Done!
- Querying repositories for available source packages ... Done!
- Downloading jsonlite from CRAN ...             OK [1 Mb in 0.24s]
Successfully downloaded 1 package in 4.7 seconds.

# Installing packages
---------------------------------------------------------
- Installing jsonlite ...                        FAILED
Error: Error installing package 'jsonlite':
====================================

* installing *source* package 'jsonlite' ...
** this is package 'jsonlite' version '1.8.9'
```

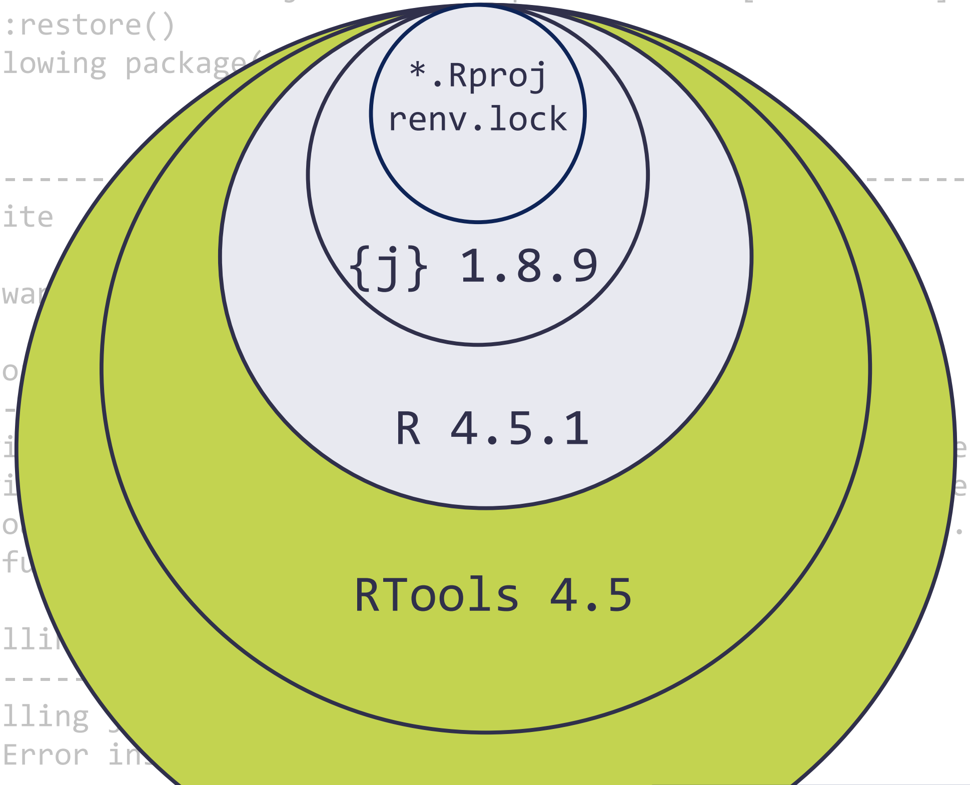




```
ℹ Using R 4.5.1 (lockfile was generated with R 4.4.2)
- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
The following package(s) will be updated:

# CRAN
-----------------------------------------------------------------------
- jsonlite   [* -> 1.8.9]

Do you want to proceed? [Y/n]: y

# Downloading packages
-------------------------------------------------------
- Querying repositories for available binary packages ... Done!
- Querying repositories for available source packages ... Done!
- Downloading jsonlite from CRAN ...            OK [1 Mb in 0.24s]
Successfully downloaded 1 package in 4.7 seconds.

# Installing packages
--------------------------------------------------------
- Installing jsonlite ...                       FAILED
Error: Error installing package 'jsonlite':
====================================

* installing *source* package 'jsonlite' ...
** this is package 'jsonlite' version '1.8.9'
```

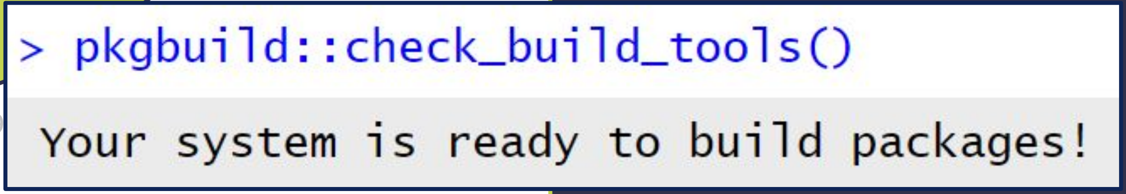
Do I have the configuration required for source compilation?

```
i Using R 4.5.1 (lockfile was generated with R 4.4.2)
- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
The following package(

# CRAN
----------------                                      -----------------
- jsonlite

Do you wa

# Downlo
--------
- Queryi                                                             e!
- Queryi                                                             e!
- Downlo                                                             .24s]
Successfu

# Installi
-----------
- Installing j
Error: Error in
==================

* installing *source* packag      ..
** this is package 'jsonlite' version '1.8.9
```



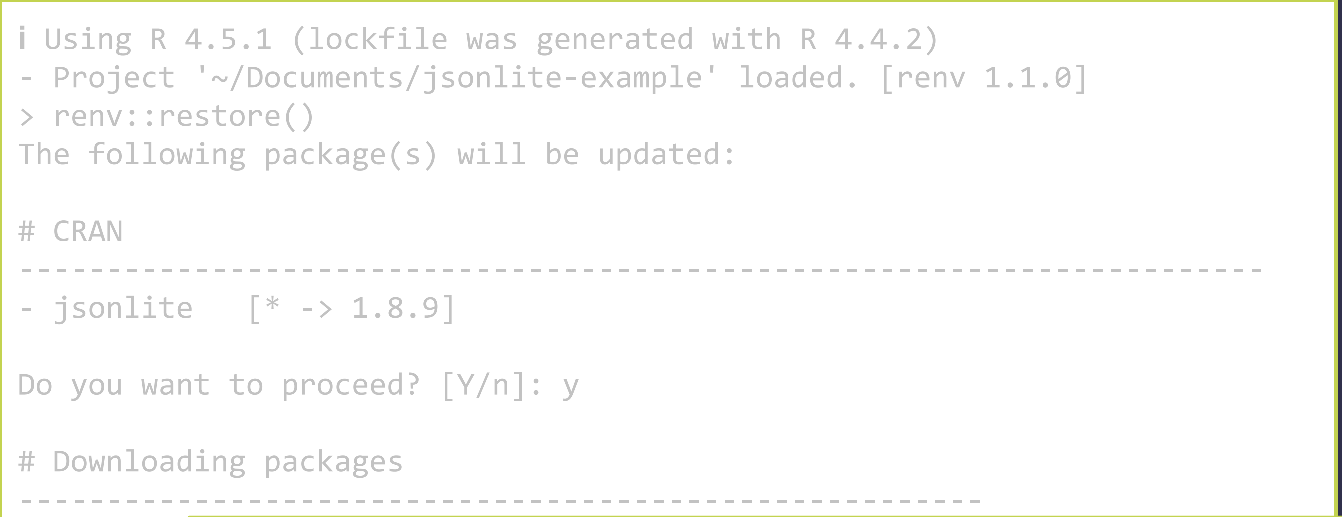
Do I have the configuration required for source compilation?

✓ yes

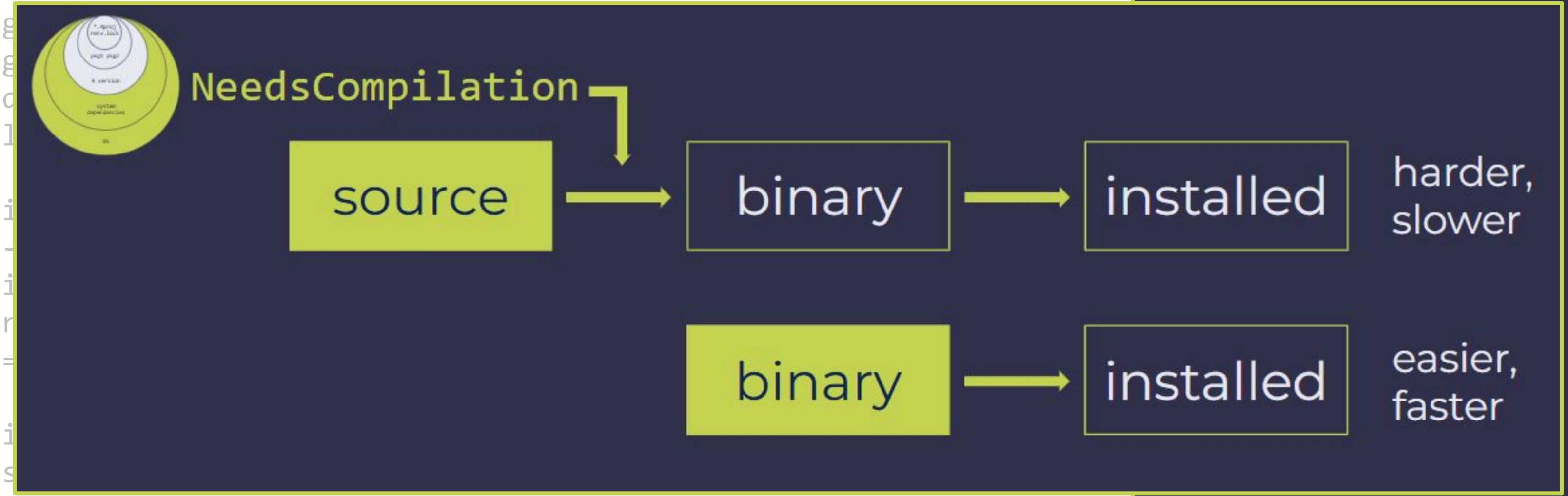


```
> pkgbuild::check_build_tools()
Your system is ready to build packages!
```

Can I install from binary instead?



```
i Using R 4.5.1 (lockfile was generated with R 4.4.2)
- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
The following package(s) will be updated:

# CRAN
-----------------------------------------------------------------------
- jsonlite   [* -> 1.8.9]

Do you want to proceed? [Y/n]: y

# Downloading packages
-------------------------------------------------------
- Querying
- Querying
- Downloac
Successful

# Installi
----------
- Installi
Error: Err
==========

* installi
** this is
```



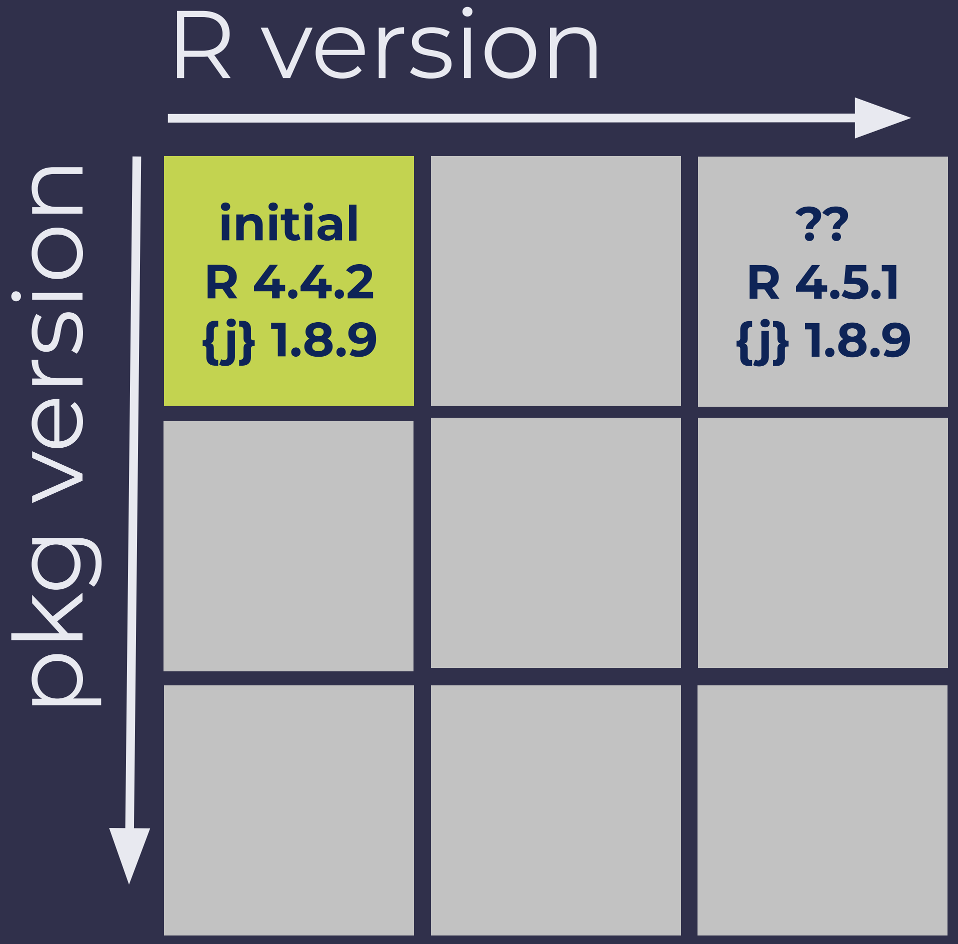
binaries (generally) available

2025-02-01

CRAN & P3M

future

CRAN: windows, macOS
P3M: windows, macOS, & linux



R version

pkg version

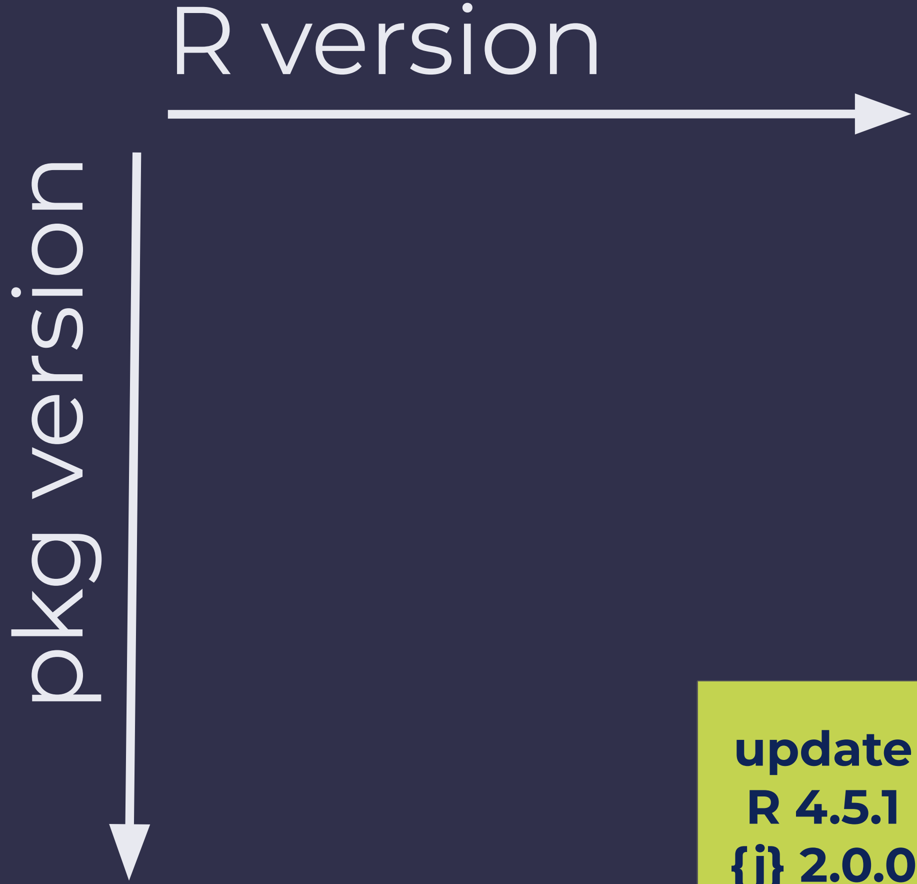
initial R 4.4.2 {j} 1.8.9		?? R 4.5.1 {j} 1.8.9

binaries (generally) available

2025-08-01

CRAN
&
P3M

CRAN: windows, macOS
P3M: windows, macOS, & linux



**update
R 4.5.1
{j} 2.0.0**

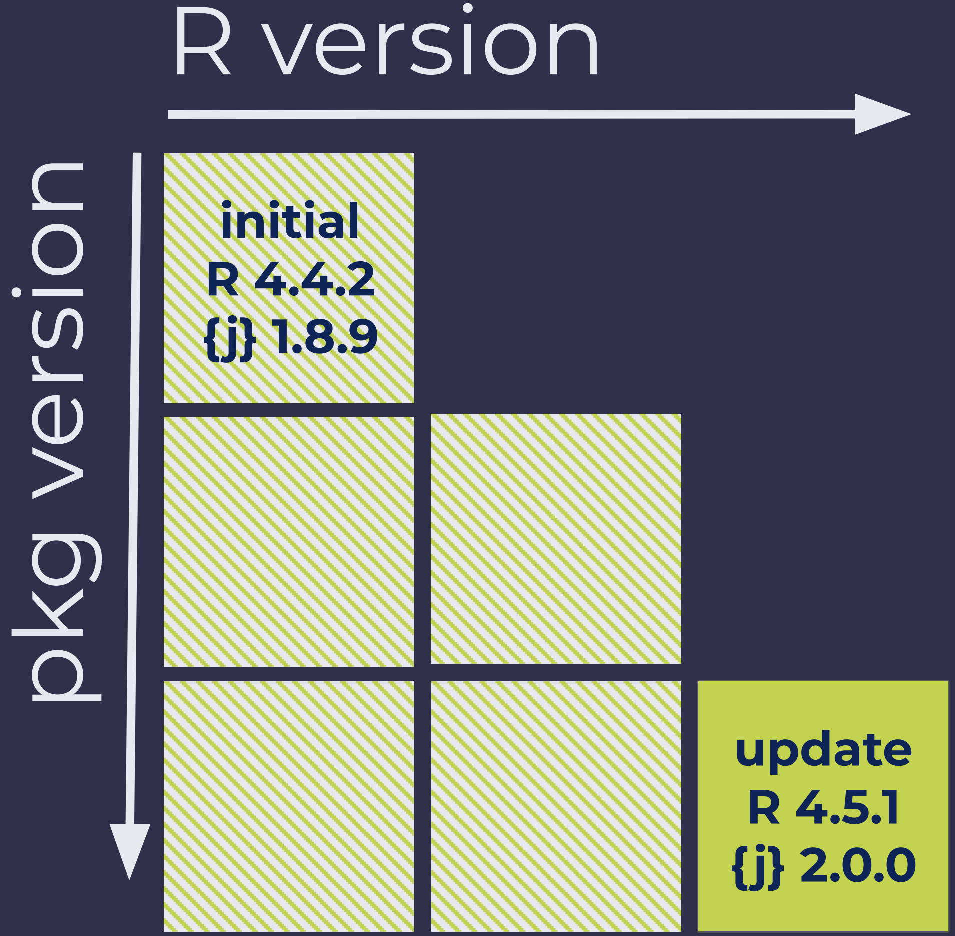
binaries (generally) available

2025-08-01

CRAN & P3M

P3M

CRAN: windows, macOS
P3M: windows, macOS, & linux



R version

pkg version

initial
R 4.4.2
{j} 1.8.9

update
R 4.5.1
{j} 2.0.0

binaries (generally) available

2025-08-01

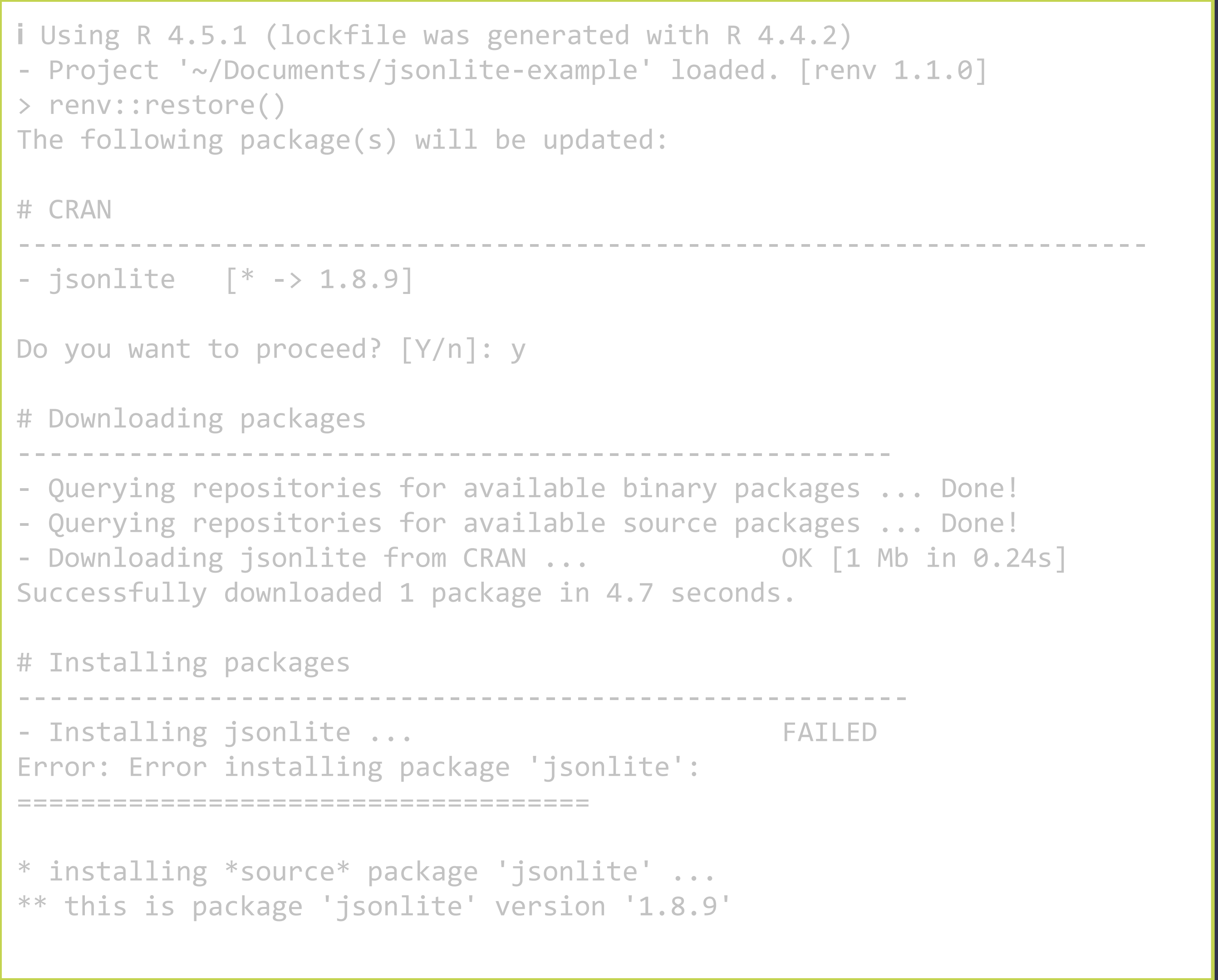
🟩	▨	□
CRAN & P3M	P3M	no

CRAN: windows, macOS
P3M: windows, macOS, & linux

R version →

pkg version ↓

▨ initial R 4.4.2 {j} 1.8.9	□	□ ?? R 4.5.1 {j} 1.8.9
▨	▨	□
▨	▨	🟩 update R 4.5.1 {j} 2.0.0



```
ℹ Using R 4.5.1 (lockfile was generated with R 4.4.2)
- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
The following package(s) will be updated:

# CRAN -----------------------------------------------------------------------
- jsonlite   [* -> 1.8.9]

Do you want to proceed? [Y/n]: y

# Downloading packages -------------------------------------------------------
- Querying repositories for available binary packages ... Done!
- Querying repositories for available source packages ... Done!
- Downloading jsonlite from CRAN ...            OK [1 Mb in 0.24s]
Successfully downloaded 1 package in 4.7 seconds.

# Installing packages --------------------------------------------------------
- Installing jsonlite ...                       FAILED
Error: Error installing package 'jsonlite':
====================================

* installing *source* package 'jsonlite' ...
** this is package 'jsonlite' version '1.8.9'
```

Can I install from binary instead?

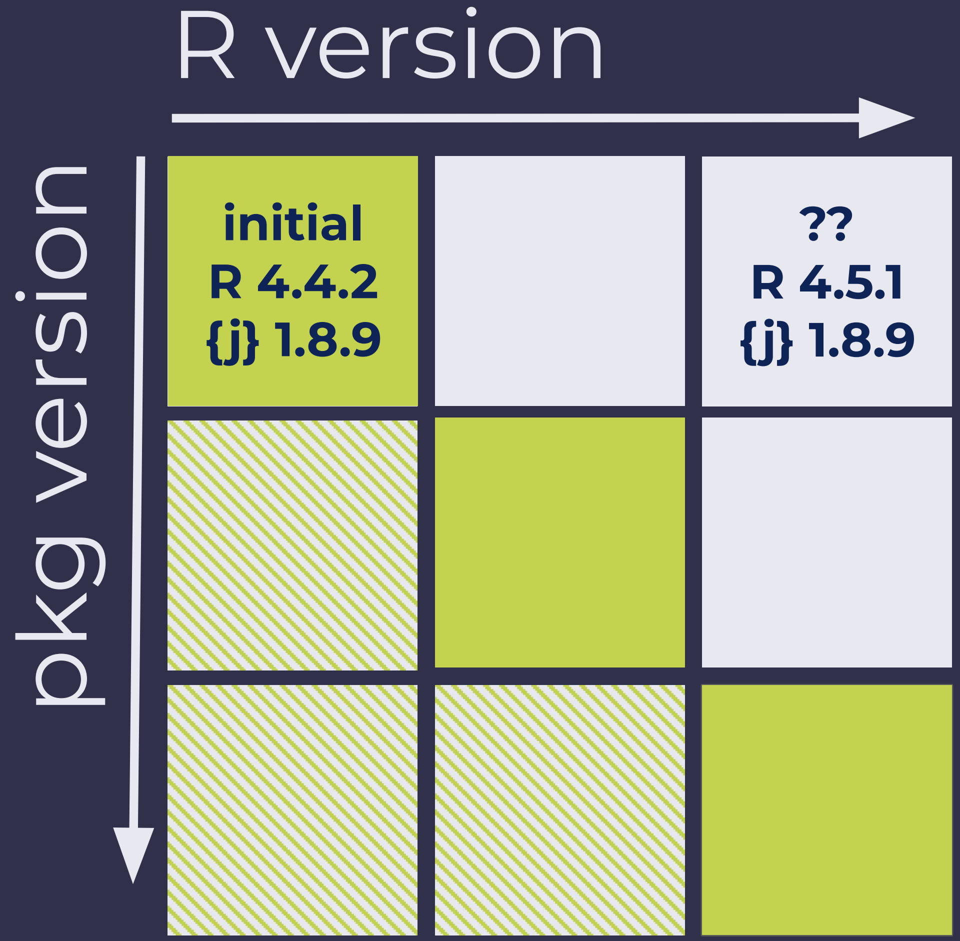
✗ no

```
ℹ Using R 4.5.1 (lockfile was generated
  with R 4.4.2)

- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
The following package(s) will be updated:

# CRAN
-----------------------------------------------------------------------
- jsonlite   [* -> 1.8.9]

Do you want to proceed? [Y/n]: y

# Downloading packages
-------------------------------------------------------
- Querying repositories for available binary packages ... Done!
- Querying repositories for available source packages ... Done!
- Downloading jsonlite from CRAN ...            OK [1 Mb in 0.24s]
Successfully downloaded 1 package in 4.7 seconds.

# Installing packages
--------------------------------------------------------
- Installing jsonlite ...                       FAILED
Error: Error installing package 'jsonlite':
====================================
```

Am I doing something weird?

R ecosystem

🟩	▨	□
yes	ok	no



R version

pkg version

🟩 **initial** **R 4.4.2** **{j} 1.8.9**	□	□ **??** **R 4.5.1** **{j} 1.8.9**
▨	🟩	□
▨	▨	🟩

R ecosystem

🟩	▨	⬜
yes	ok	no

R version →

pkg version ↓

🟩 **initial** **R 4.4.2** **{j} 1.8.9**	⬜	⬜ **??** **R 4.5.1** **{j} 1.8.9**
▨	🟩	⬜
▨	▨	🟩

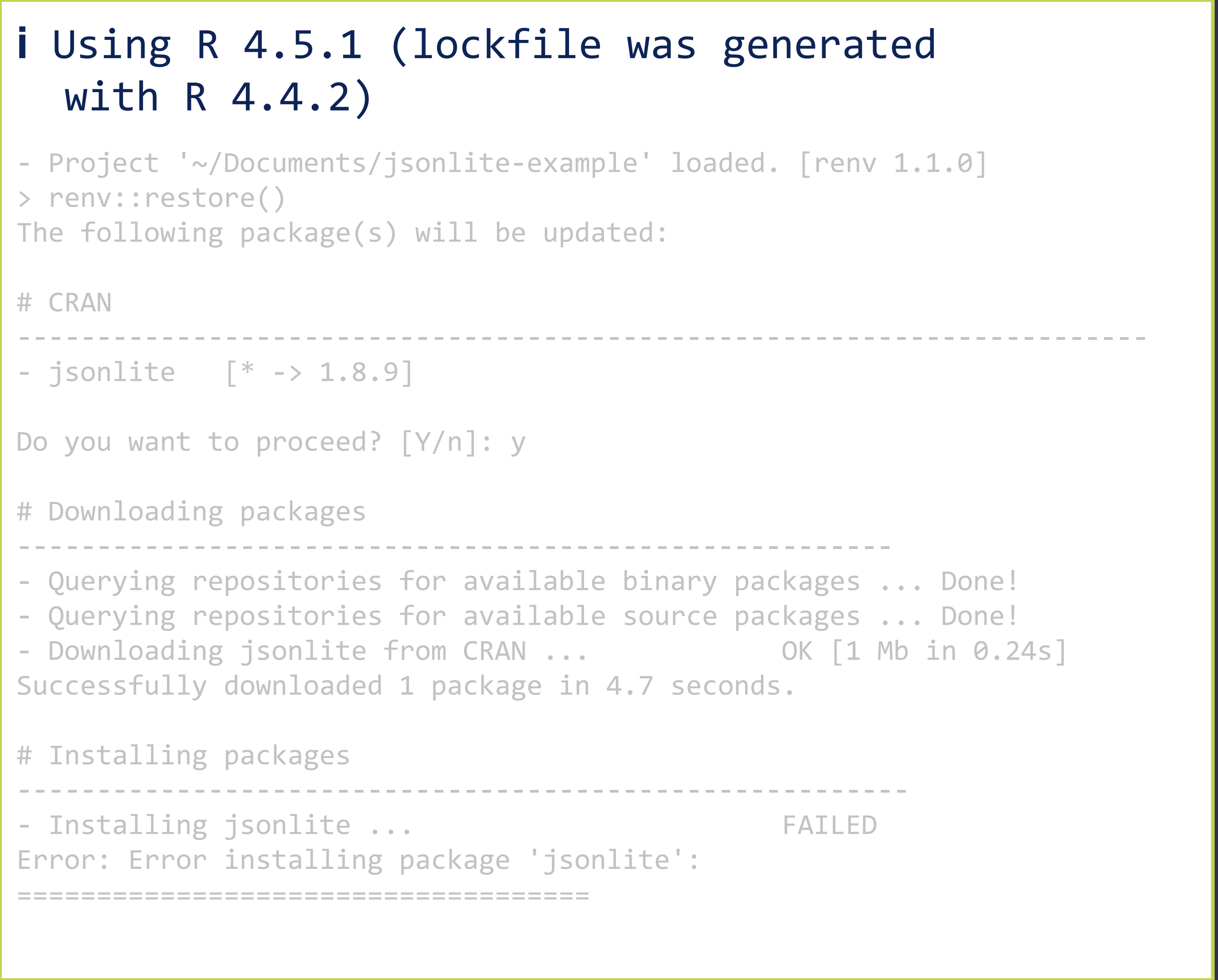
R ecosystem

🟩 yes	▨ ok	⬜ no

R version →

pkg version ↓

🟩 initial R 4.4.2 {j} 1.8.9	⬜	⬜ ?? R 4.5.1 {j} 1.8.9
▨	🟩	⬜
▨	▨	🟩



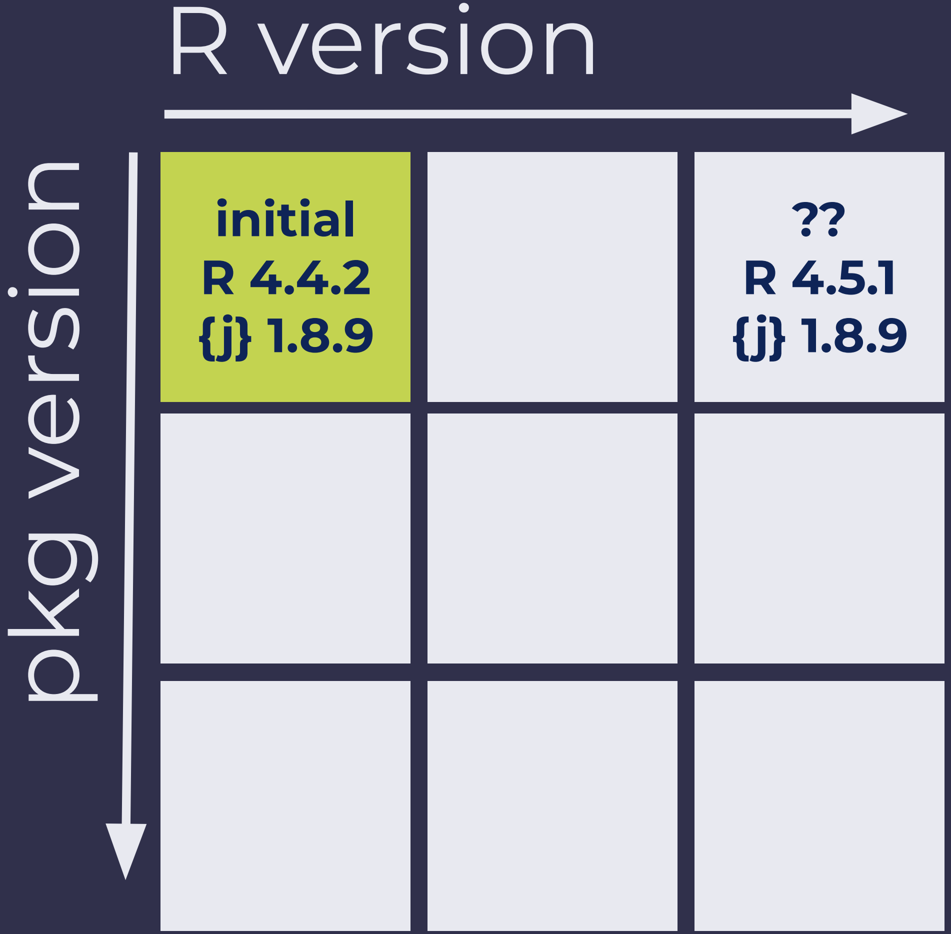
ℹ Using R 4.5.1 (lockfile was generated with R 4.4.2)

```
- Project '~/Documents/jsonlite-example' loaded. [renv 1.1.0]
> renv::restore()
The following package(s) will be updated:

# CRAN
-----------------------------------------------------------------------
- jsonlite   [* -> 1.8.9]

Do you want to proceed? [Y/n]: y

# Downloading packages
-------------------------------------------------------
- Querying repositories for available binary packages ... Done!
- Querying repositories for available source packages ... Done!
- Downloading jsonlite from CRAN ...            OK [1 Mb in 0.24s]
Successfully downloaded 1 package in 4.7 seconds.

# Installing packages
--------------------------------------------------------
- Installing jsonlite ...                       FAILED
Error: Error installing package 'jsonlite':
====================================
```

Am I doing something weird?

✓ yes

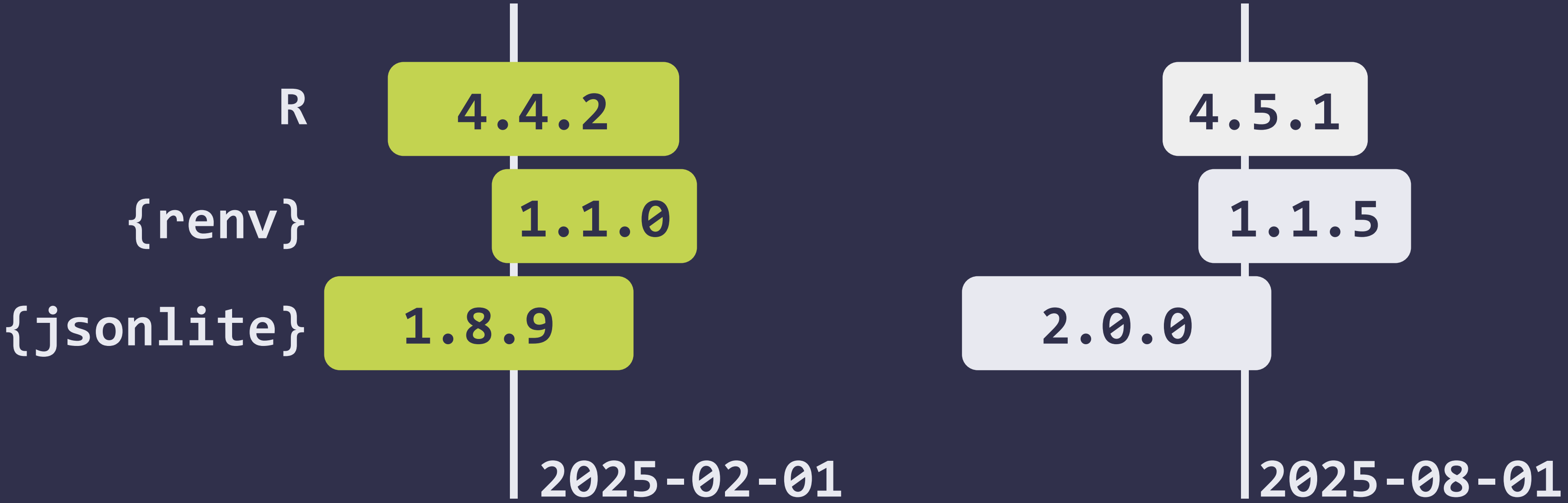
“freeze” a project by returning to its initial state



initial R 4.4.2 {j} 1.8.9		?? R 4.5.1 {j} 1.8.9

resume project

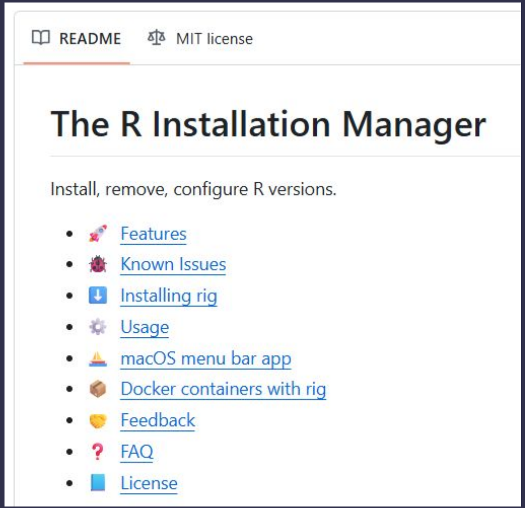
1 freeze packages



	2025-02-01	2025-08-01
R	4.4.2	4.5.1
{renv}	1.1.0	1.1.5
{jsonlite}	1.8.9	2.0.0

use rig to switch between R versions

https://github.com/r-lib/rig



The R Installation Manager

Install, remove, configure R versions.

- 🚀 Features
- 🐞 Known Issues
- ⬇️ Installing rig
- ⚙️ Usage
- ⛵ macOS menu bar app
- 📦 Docker containers with rig
- 🤝 Feedback
- ❓ FAQ
- 📘 License

terminal

```
> rig default 4.4.2
```

resume project

1 freeze packages

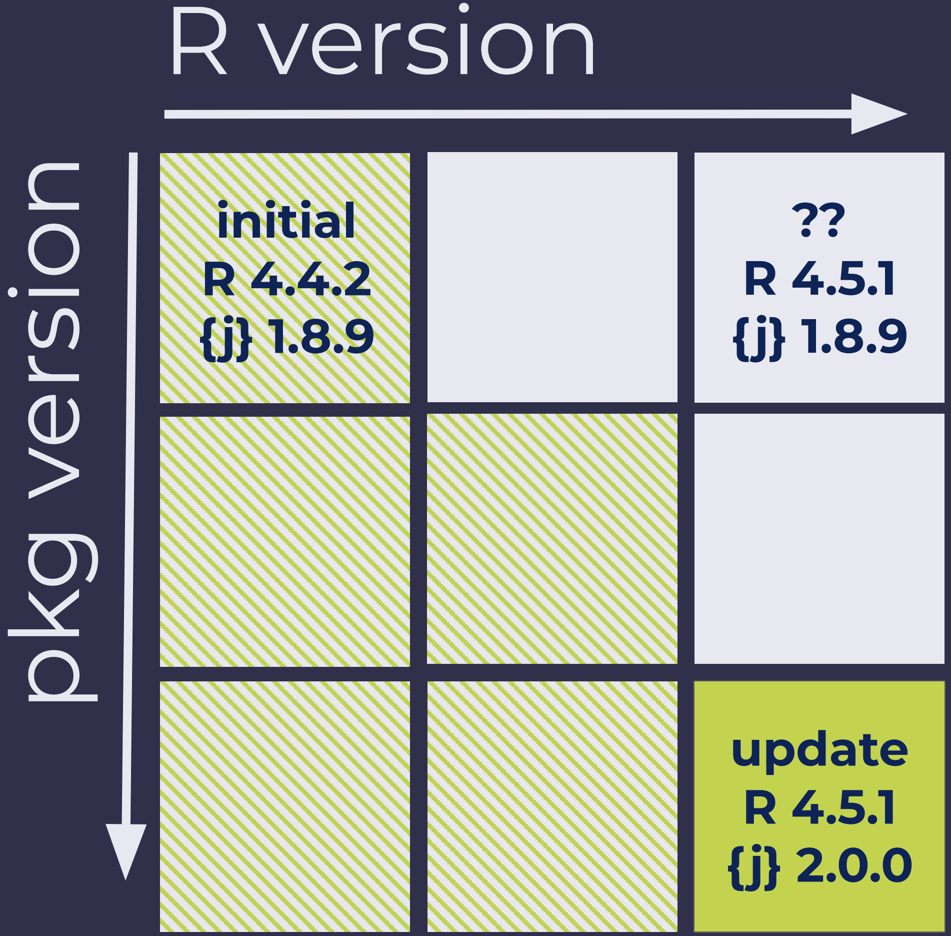
```
R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
- Project '~/jsonlite-example' loaded. [renv 1.1.0]

# install package versions from repository specified in lockfile
> renv::restore()
```

binaries (generally) available

2025-08-01

🟩	▨	□
CRAN & P3M	P3M	no

CRAN: Windows, mac OS
P3M: Windows, mac OS, & linux



pkg version ↓ / R version →		
▨ initial R 4.4.2 {j} 1.8.9	□	□ ?? R 4.5.1 {j} 1.8.9
▨	▨	□
▨	▨	🟩 update R 4.5.1 {j} 2.0.0

resume project

1 freeze packages

```
R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
- Project '~/jsonlite-example' loaded. [renv 1.1.0]

# install package versions from repository specified in lockfile
> renv::restore()

# install package versions from P3M
> renv::restore(
     repos = c("P3M" = "https://packagemanager.posit.co/cran/latest")
  )
```

resume project

1 freeze packages

```
R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
- Project '~/jsonlite-example' loaded. [renv 1.1.0]

# install package versions from repository specified in lockfile
> renv::restore()

# install package versions from P3M
> renv::restore(
    repos = c("P3M" = "https://packagemanager.posit.co/cran/latest")
  )

# install date-based package versions from P3M
> renv::checkout(date = "2025-02-01")
```

project decisions

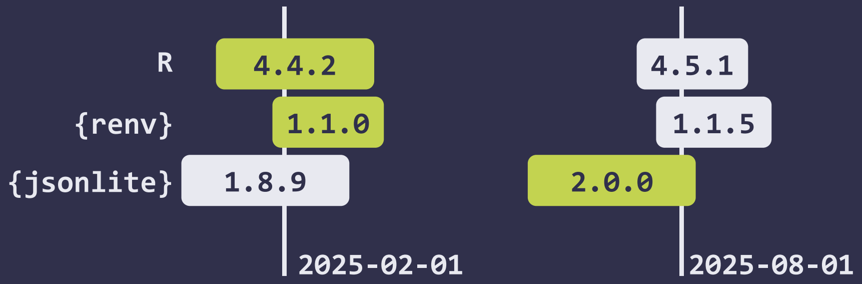
1 freeze

2 manage

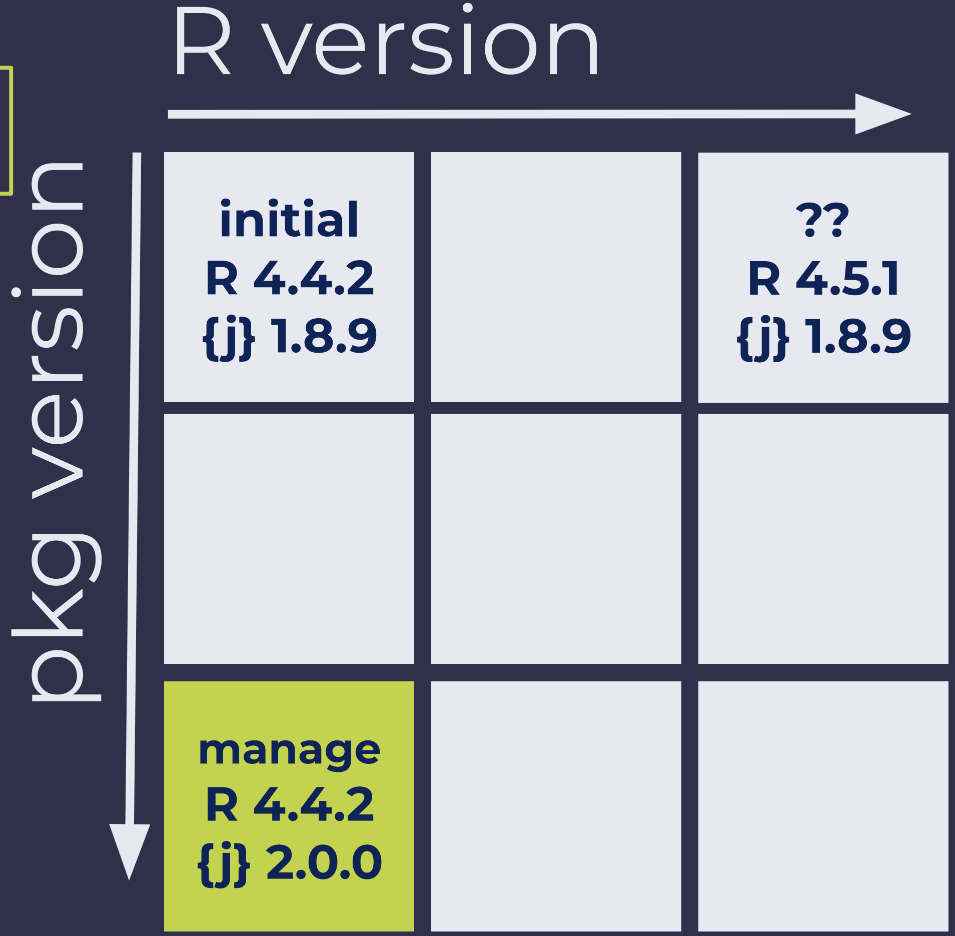
3 update

resume project

2 manage packages



	2025-02-01	2025-08-01
R	4.4.2	4.5.1
{renv}	1.1.0	1.1.5
{jsonlite}	1.8.9	2.0.0



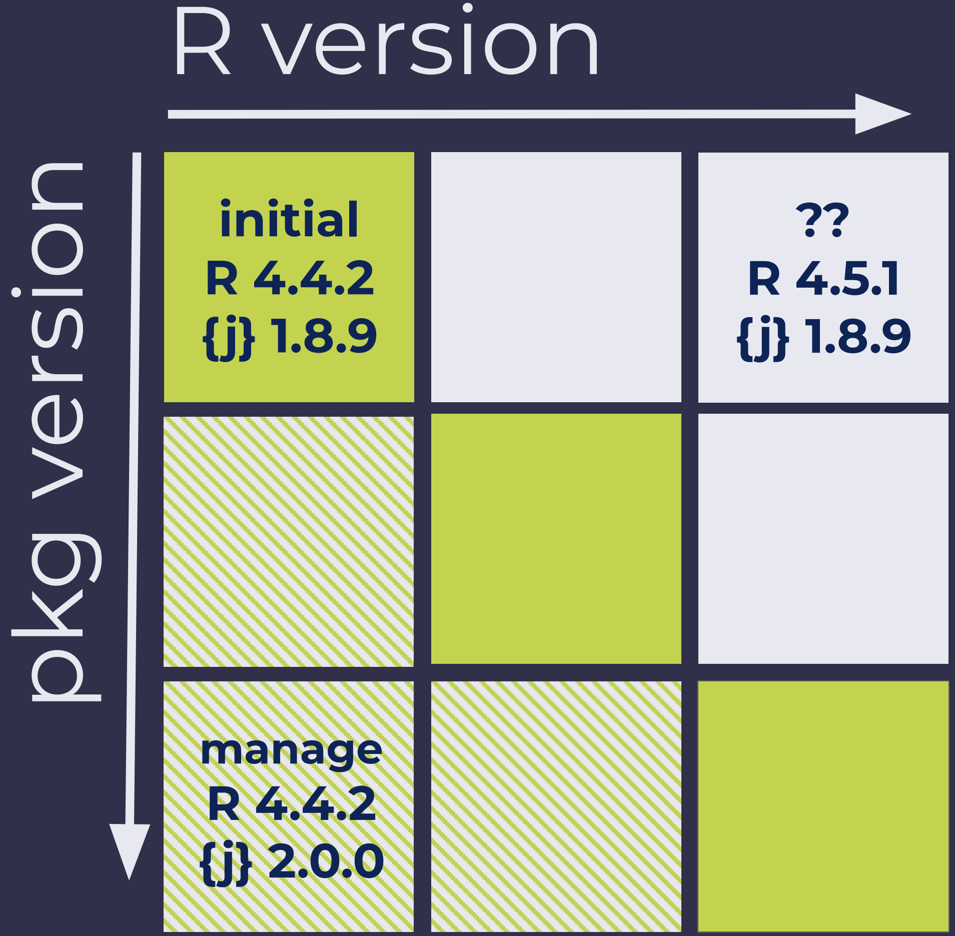
R version

pkg version

initial R 4.4.2 {j} 1.8.9		?? R 4.5.1 {j} 1.8.9
manage R 4.4.2 {j} 2.0.0		

R ecosystem

🟩	▨	⬜
yes	ok	no



R version

pkg version

🟩 initial R 4.4.2 {j} 1.8.9	⬜	⬜ ?? R 4.5.1 {j} 1.8.9
▨	🟩	⬜
▨ manage R 4.4.2 {j} 2.0.0	▨	🟩

resume project

2 manage packages

```
R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
- Project '~/jsonlite-example' loaded. [renv 1.1.0]


# install latest package version from repository specified in lockfile
> renv::install("jsonlite")
> renv::snapshot()
```

project decisions

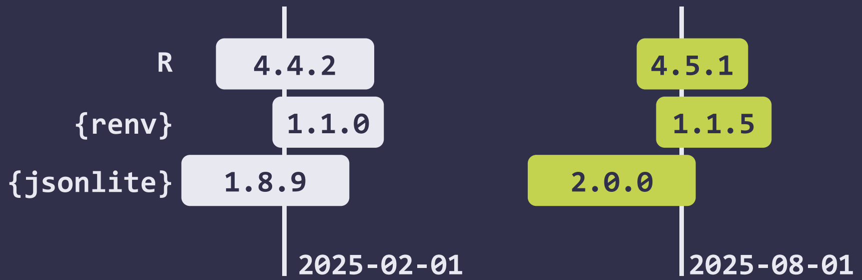
1 freeze

2 manage

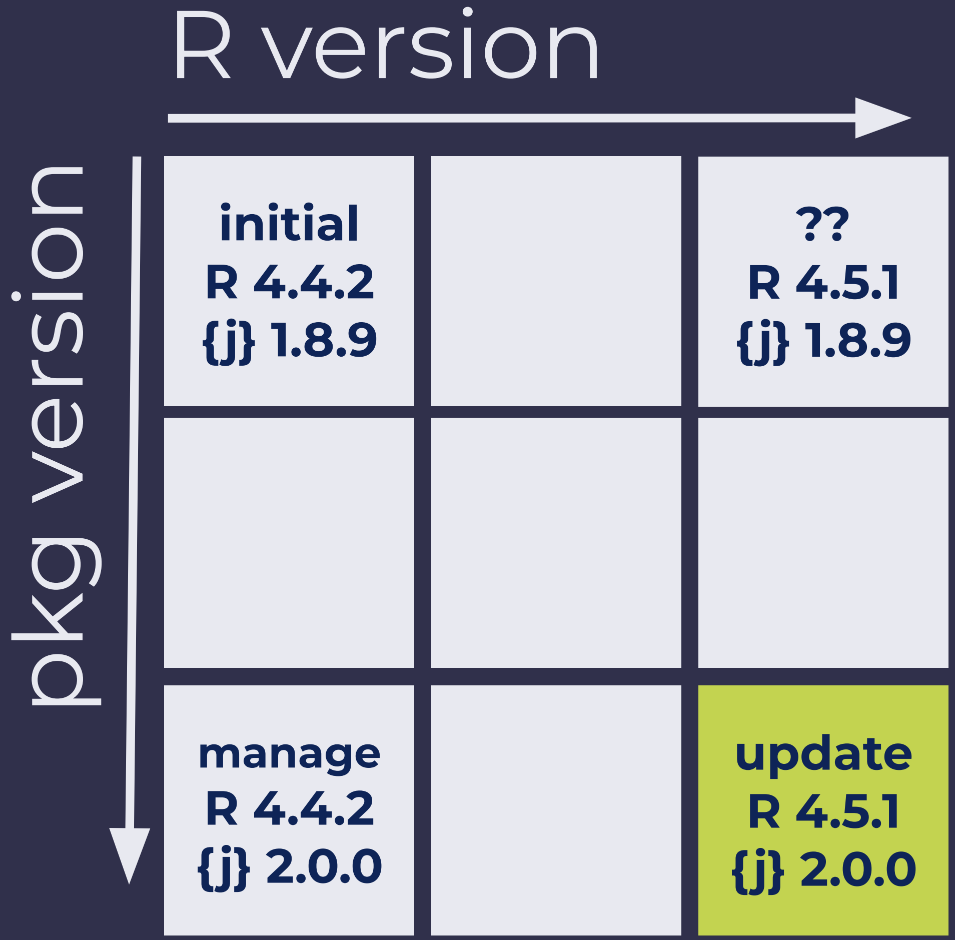
3 update

resume project

3 update packages



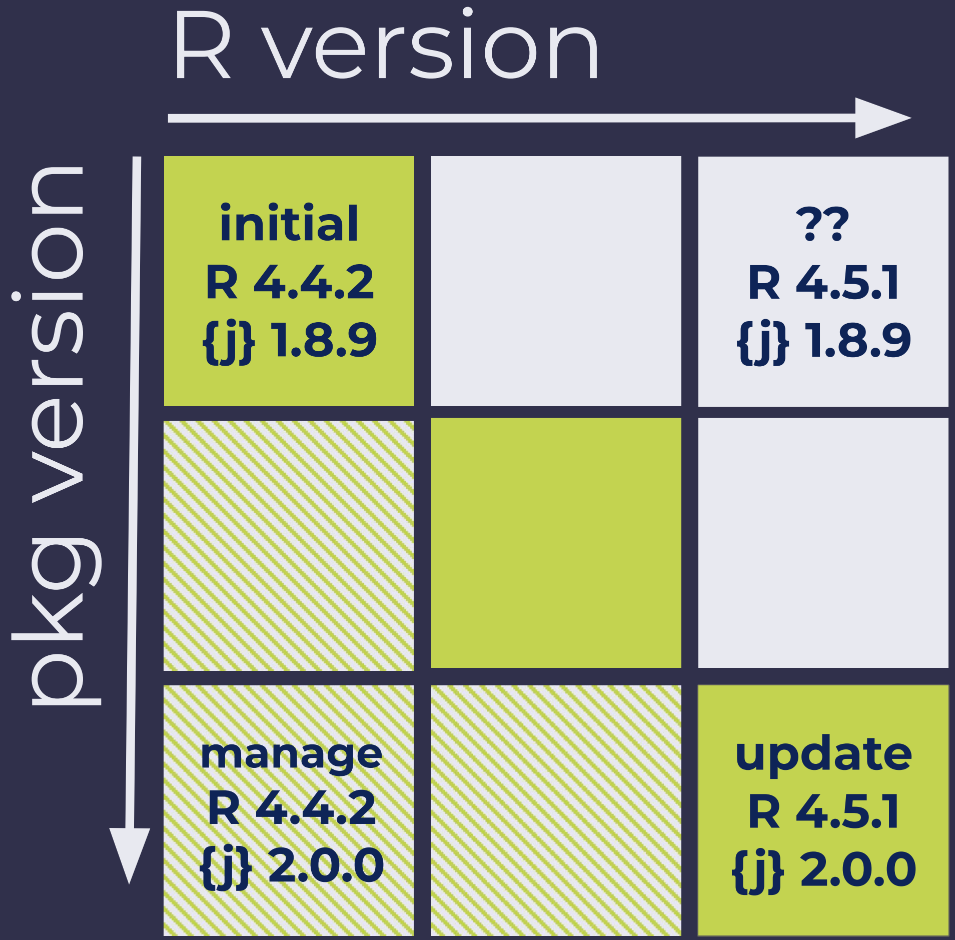
	2025-02-01	2025-08-01
R	4.4.2	4.5.1
{renv}	1.1.0	1.1.5
{jsonlite}	1.8.9	2.0.0



initial R 4.4.2 {j} 1.8.9		?? R 4.5.1 {j} 1.8.9
manage R 4.4.2 {j} 2.0.0		update R 4.5.1 {j} 2.0.0

R ecosystem

🟩 yes	▨ ok	⬜ no



🟩 initial R 4.4.2 {j} 1.8.9	⬜	⬜ ?? R 4.5.1 {j} 1.8.9
▨	🟩	⬜
▨ manage R 4.4.2 {j} 2.0.0	▨	🟩 update R 4.5.1 {j} 2.0.0

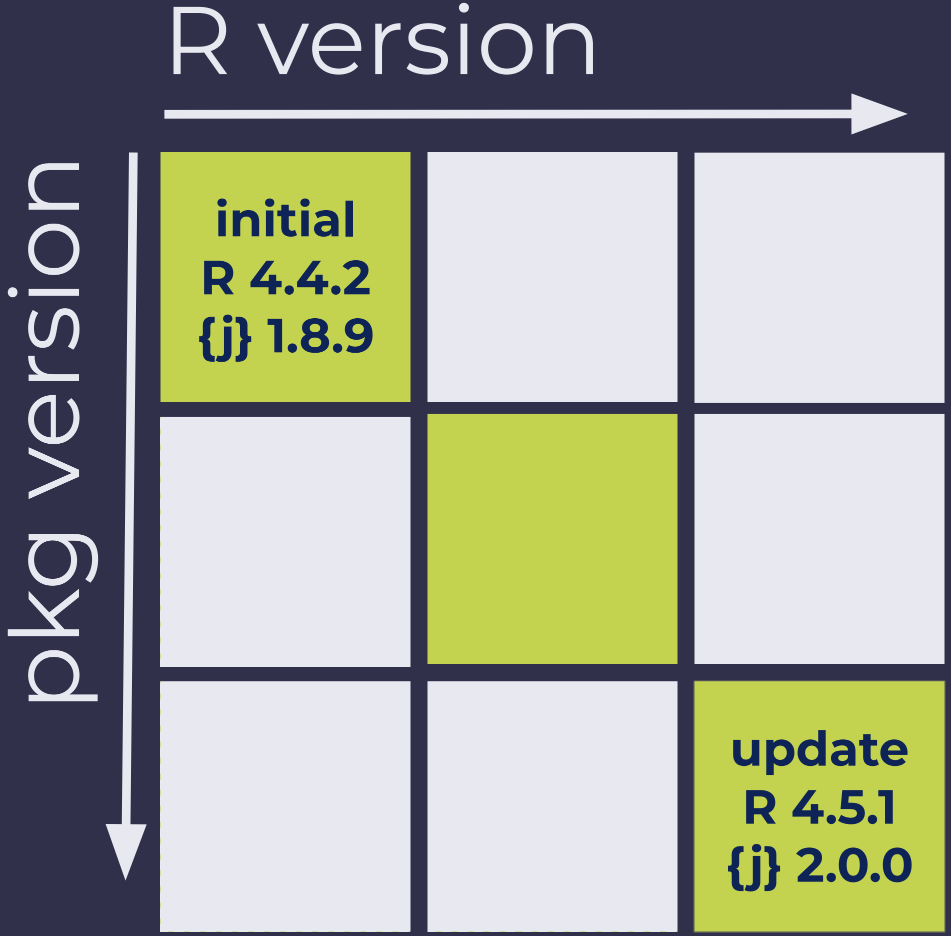
resume project

3 update packages

```
ℹ Using R 4.5.1 (lockfile was generated with R 4.4.2)
- Project '~/jsonlite-example' loaded. [renv 1.1.0]

# use latest version of renv
> renv::upgrade()

# install all latest package versions
> renv::install()

# record changes to both R version and package versions in lockfile
> renv::snapshot()
```

for active projects, regularly updating is the happy path

(+ git version control)



R version

pkg version

initial
R 4.4.2
{j} 1.8.9

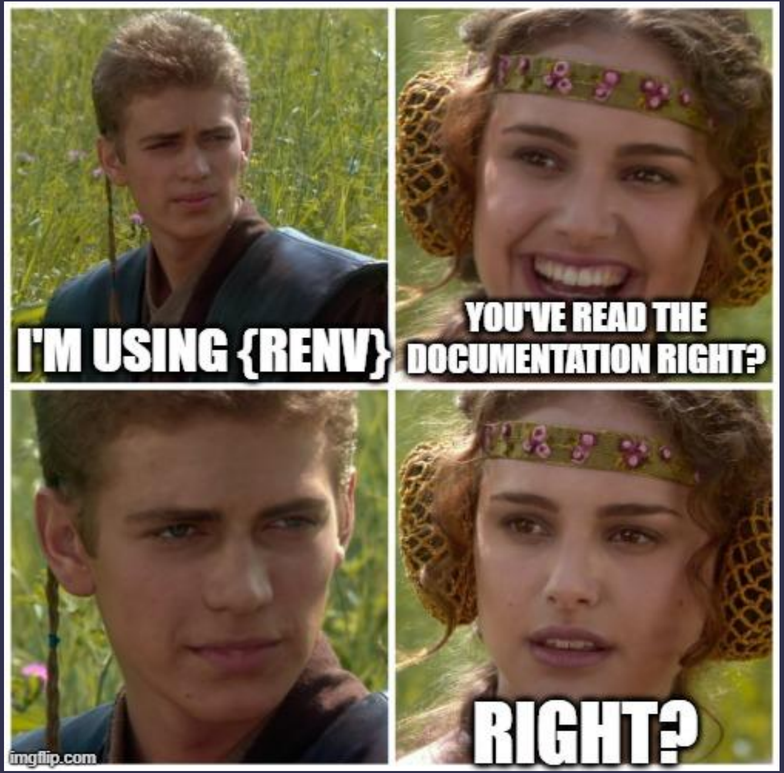
update
R 4.5.1
{j} 2.0.0

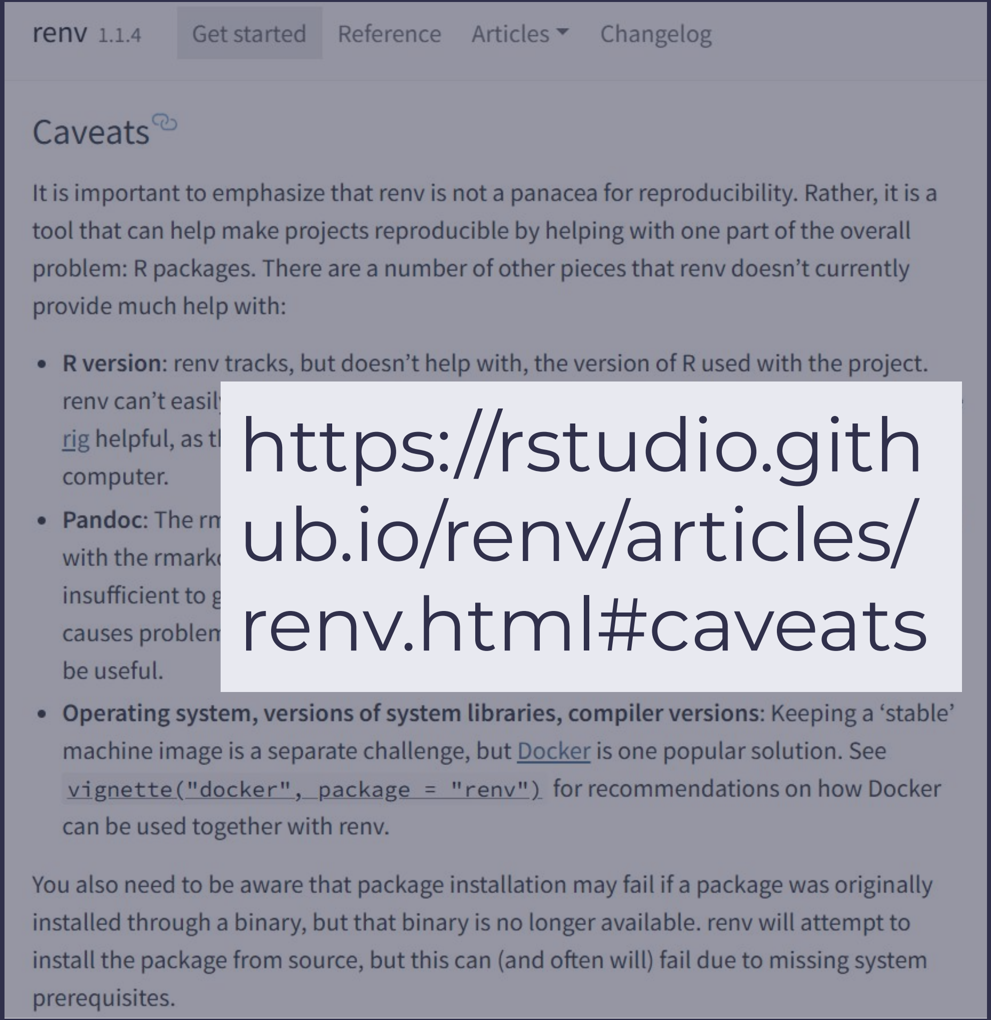
intro to
{renv}

package
installation

projects
over time

tips





Caveats

It is important to emphasize that renv is not a panacea for reproducibility. Rather, it is a tool that can help make projects reproducible by helping with one part of the overall problem: R packages. There are a number of other pieces that renv doesn't currently provide much help with:

- **R version**: renv tracks, but doesn't help with, the version of R used with the project. renv can't easil
rig helpful, as t
computer.
- **Pandoc**: The rm
with the rmark
insufficient to g
causes problem
be useful.
- **Operating system, versions of system libraries, compiler versions**: Keeping a 'stable' machine image is a separate challenge, but Docker is one popular solution. See `vignette("docker", package = "renv")` for recommendations on how Docker can be used together with renv.

You also need to be aware that package installation may fail if a package was originally installed through a binary, but that binary is no longer available. renv will attempt to install the package from source, but this can (and often will) fail due to missing system prerequisites.

https://rstudio.github.io/renv/articles/renv.html#caveats

{renv} decisions for active projects

1

freeze

short term
sustainable

2

manage

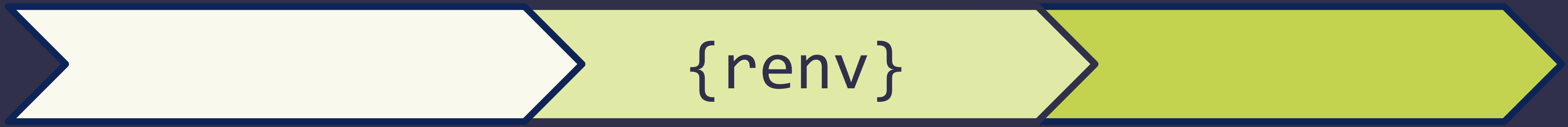
short term
sustainable

3

update

long term
sustainable

reproducibility spectrum



code

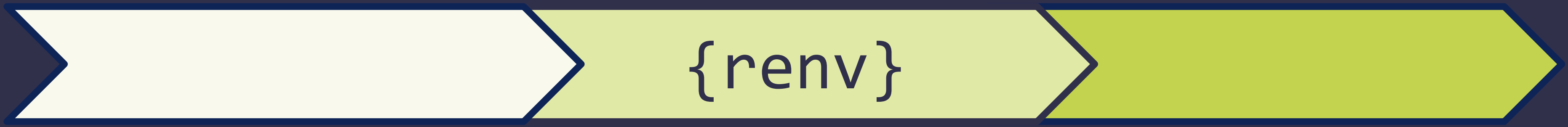
code
pkg version

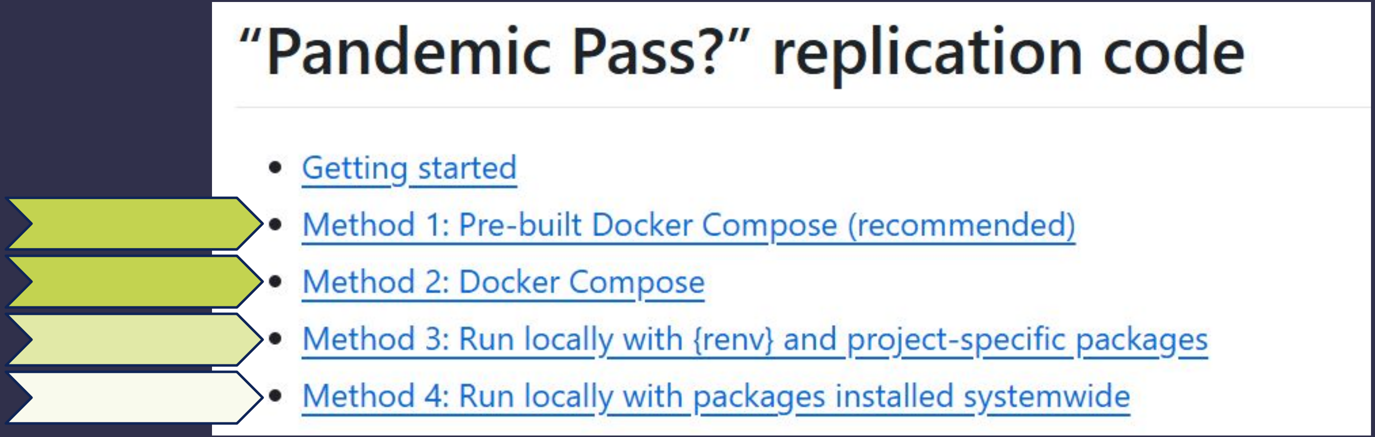
code
pkg version
R version
sys. dep.
OS

completed project example

https://github.com/andrewheiss/mountainous-mackerel-docker

reproducibility spectrum



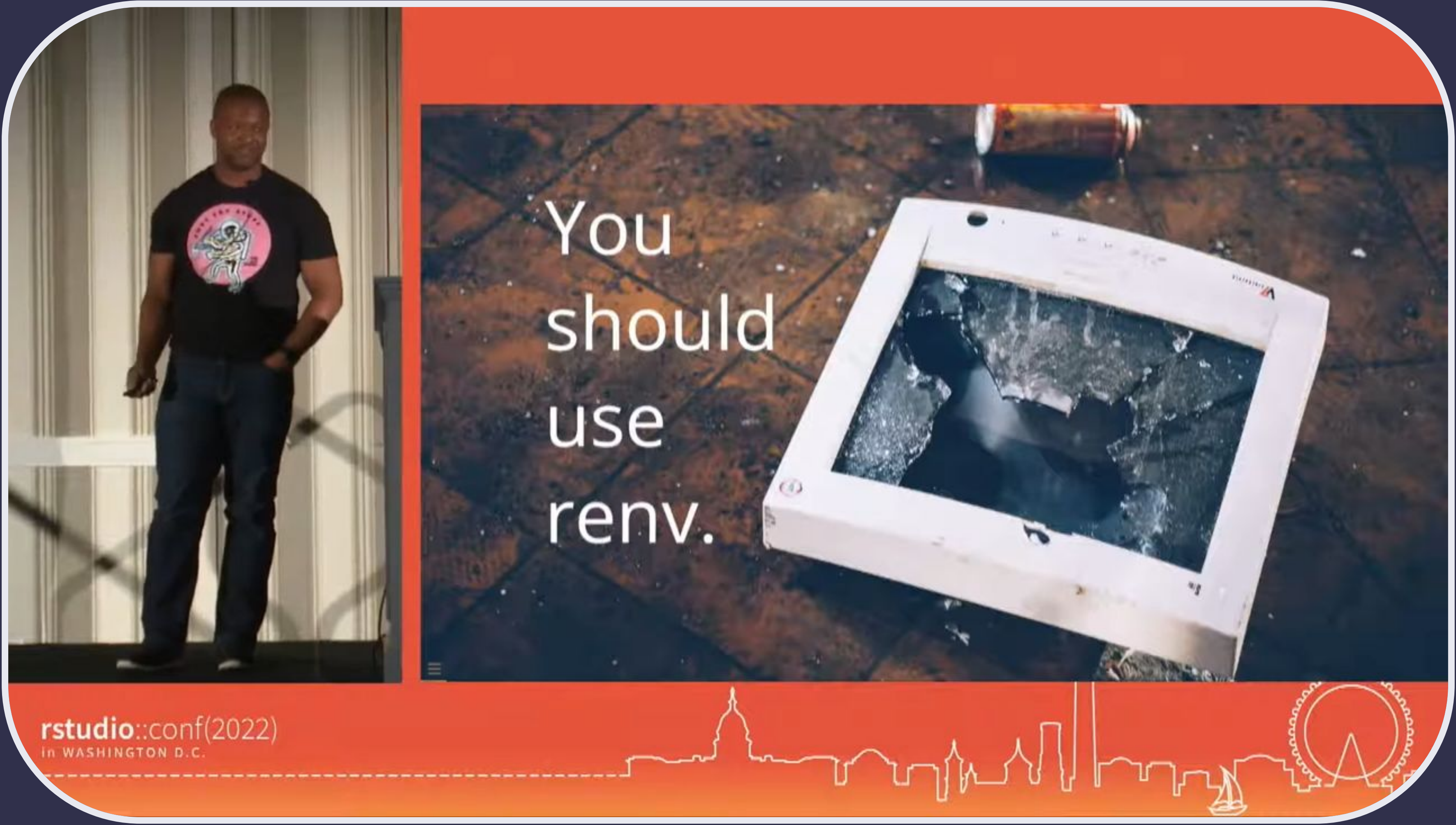


“Pandemic Pass?” replication code

- Getting started
- Method 1: Pre-built Docker Compose (recommended)
- Method 2: Docker Compose
- Method 3: Run locally with {renv} and project-specific packages
- Method 4: Run locally with packages installed systemwide

completed project example

https://github.com/andrewheiss/mountainous-mackerel-docker



You should use renv.

rstudio::conf(2022)
in WASHINGTON D.C.

You should use renv with intention.

- ★ E. David Aja
- ★ Kevin Ushey
- ★ more friends and colleagues, you know who you are!

more tips & details

https://github.com/shannonpileggi/practical-renv