Reproducing package environments

using renv to lock packages

2021-06-04

Once upon a time (6 months ago), a hero (you) wrote some code

```
add_rownames(data, var = "rowname")
```

But then!

Updates were made to the dplyr package:

```
Warning message:
`add_rownames()` was deprecated in dplyr 1.0.0.
Please use `tibble::rownames_to_column()` instead.
```

renv

renv **is designed to improve project-level reproducibility**

records and restores the packages used in a project

Successor to packrat

How does renv help?

1. **Each project gets it's own library (isolated)**

2. **The project library can be shipped with a self-contained lockfile,** renv.lock **(portable)**

3. renv.lock **can be restored with** renv::restore() **(reproducible)**

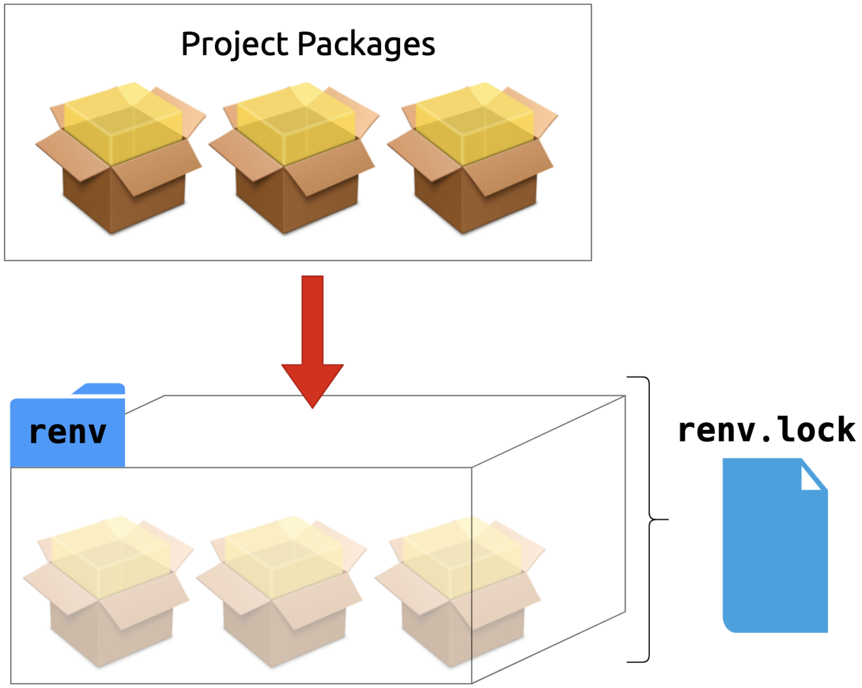
renv::init()

Creates local renv environment; caches packages.

renv::init()

Creates local renv environment; caches packages.

Documents packages in renv.lock



renv::dependencies()

```r
library(ggplot2)

targets::tar_target()

require(dplyr)

requireNamespace("devtools")
```

renv::dependencies()

```
library(ggplot2)

targets::tar_target()

require(dplyr)

requireNamespace("devtools")
```

Your Turn 1

Create a new project

Create a new file called plot.R **In that file, load** ggplot2 **and create a plot with** quickplot(mpg$displ)**.**

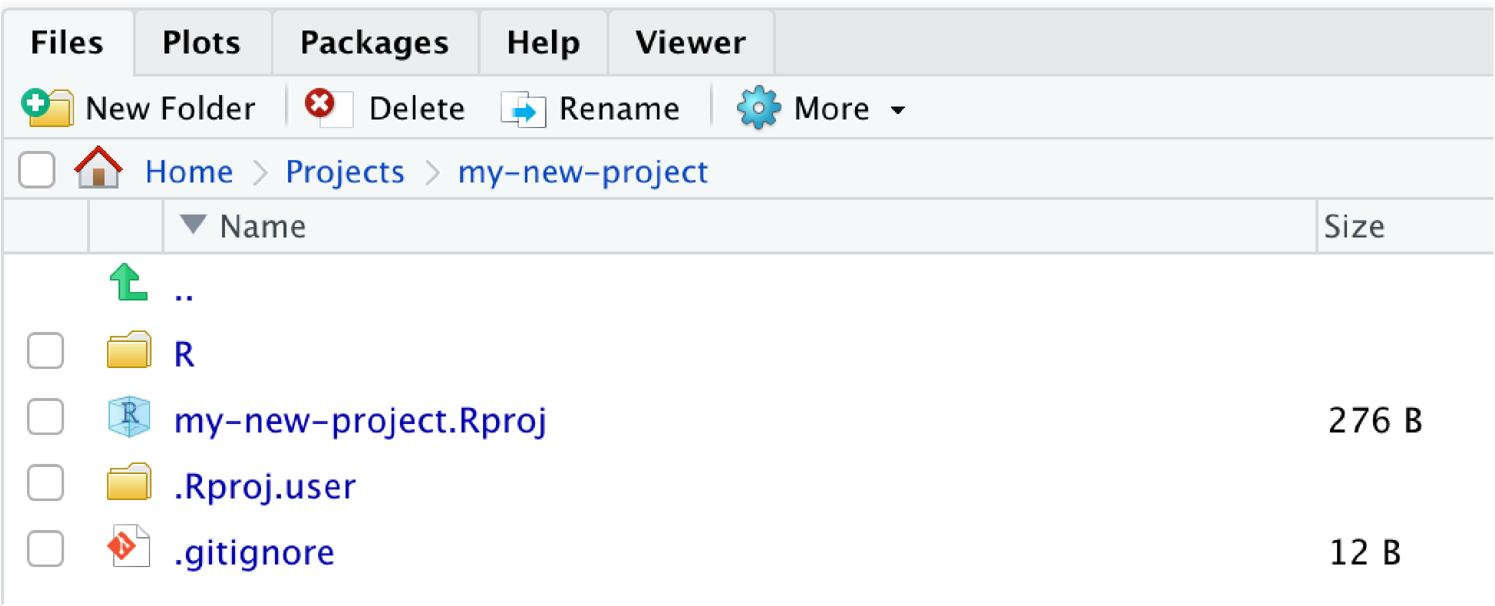
What packages does your project depend on? Make a prediction, then run renv::dependencies() **in the console (not in** plot.R**!) to see if you were right.**

In the console, initiate a renv **environment with** renv::init()

Open the file called renv.lock**. What is this information?**

Your Turn 1

In RStudio: File > New Project > 'my-new-project'



Name	Size
..	
R	
my-new-project.Rproj	276 B
.Rproj.user	
.gitignore	12 B

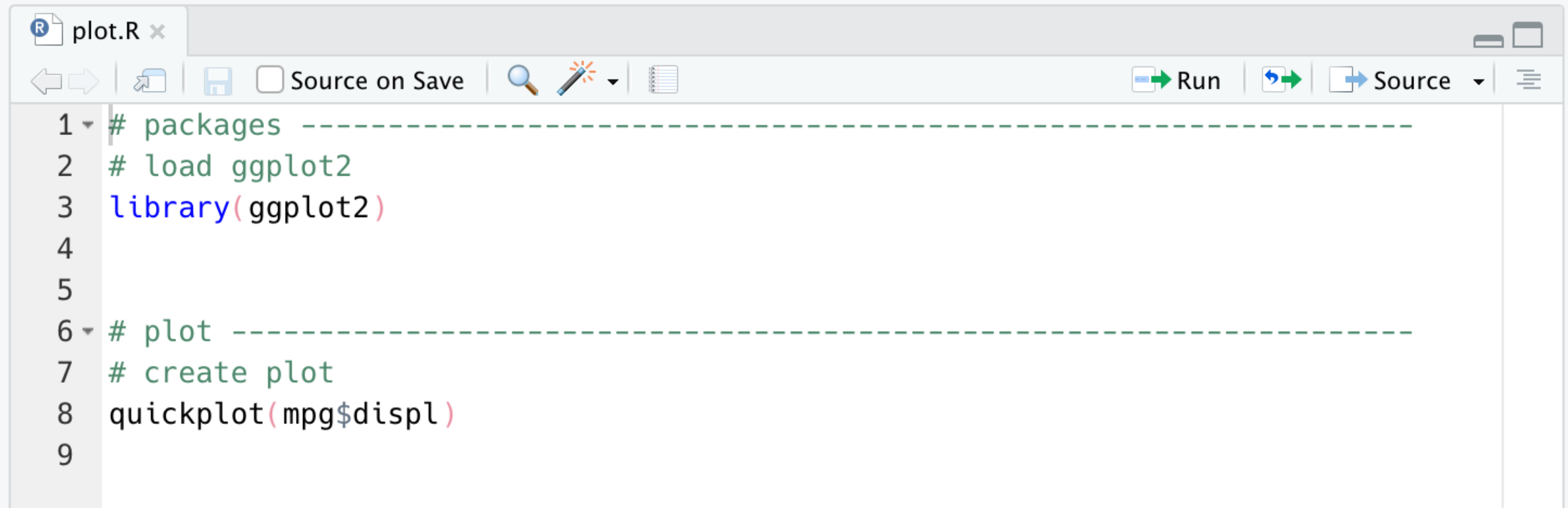
Your Turn 1

Or with usethis

```
usethis::create_project("~/Projects/my-new-project") #<<

✓ Creating '~/Projects/my-new-project/'
✓ Setting active project to '~/Projects/my-new-project'
✓ Creating 'R/'
✓ Writing 'my-new-project.Rproj'
✓ Adding '.Rproj.user' to '.gitignore'
✓ Opening '~/Projects/my-new-project/' in new RStudio session
✓ Setting active project to 'my-last-project'
```

Your Turn 1

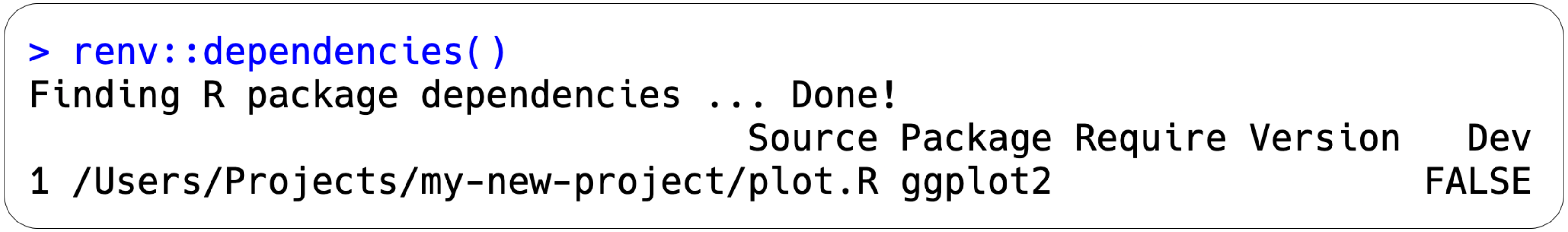


```
# packages ---------------------------------------------------------------------
# load ggplot2
library(ggplot2)


# plot -------------------------------------------------------------------------
# create plot
quickplot(mpg$displ)

```

Your Turn 1



```
> renv::dependencies()
Finding R package dependencies ... Done!
                                 Source Package Require Version   Dev
1 /Users/Projects/my-new-project/plot.R ggplot2                 FALSE
```

Your Turn 1

```
> renv::init()
* Initializing project ...
* Discovering package dependencies ... Done!
* Copying packages into the cache ... Done!
The following package(s) will be updated in the lockfile:
```

Your Turn 1

Packages in project library

```
# CRAN ===============================
- MASS           [* -> 7.3-54]
- Matrix         [* -> 1.3-3]
- R6             [* -> 2.5.0]
- RColorBrewer   [* -> 1.1-2]
- cli            [* -> 2.5.0]
- colorspace     [* -> 2.0-1]
- crayon         [* -> 1.4.1]
            ...<omitted>...
```

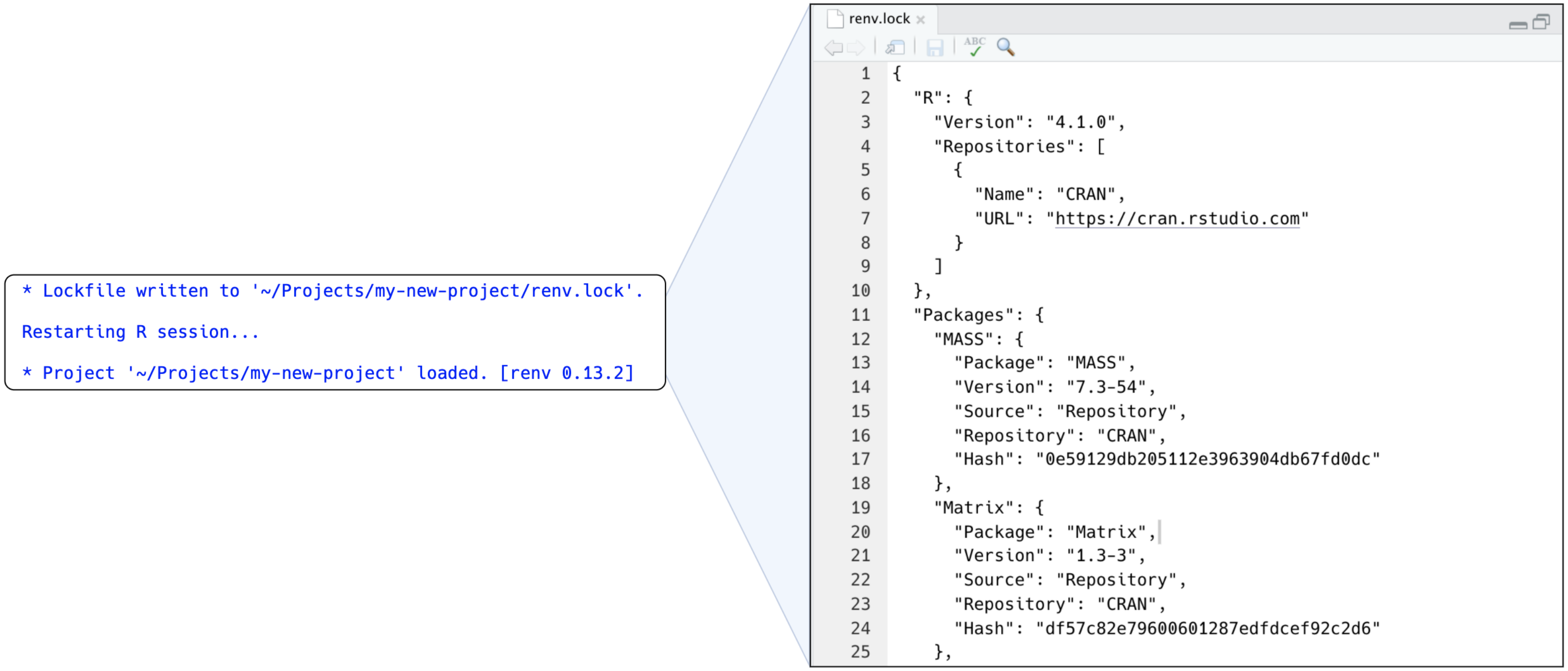
Your Turn 1

```
* Lockfile written to '~/Projects/my-new-project/renv.lock'.

Restarting R session...

* Project '~/Projects/my-new-project' loaded. [renv 0.13.2]
```

Your Turn 1



```
* Lockfile written to '~/Projects/my-new-project/renv.lock'.

Restarting R session...

* Project '~/Projects/my-new-project' loaded. [renv 0.13.2]
```

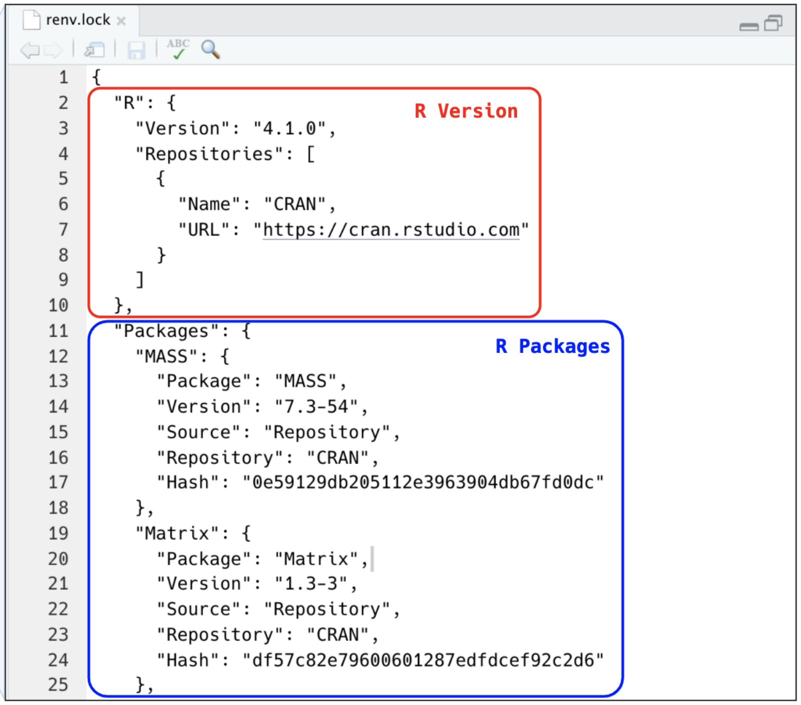
```
{
  "R": {
    "Version": "4.1.0",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cran.rstudio.com"
      }
    ]
  },
  "Packages": {
    "MASS": {
      "Package": "MASS",
      "Version": "7.3-54",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash": "0e59129db205112e3963904db67fd0dc"
    },
    "Matrix": {
      "Package": "Matrix",
      "Version": "1.3-3",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash": "df57c82e79600601287edfdcef92c2d6"
    },
```

Your Turn 1

```
* Lockfile written to '~/Projects/my-new-project/renv.lock'.

Restarting R session...

* Project '~/Projects/my-new-project' loaded. [renv 0.13.2]
```

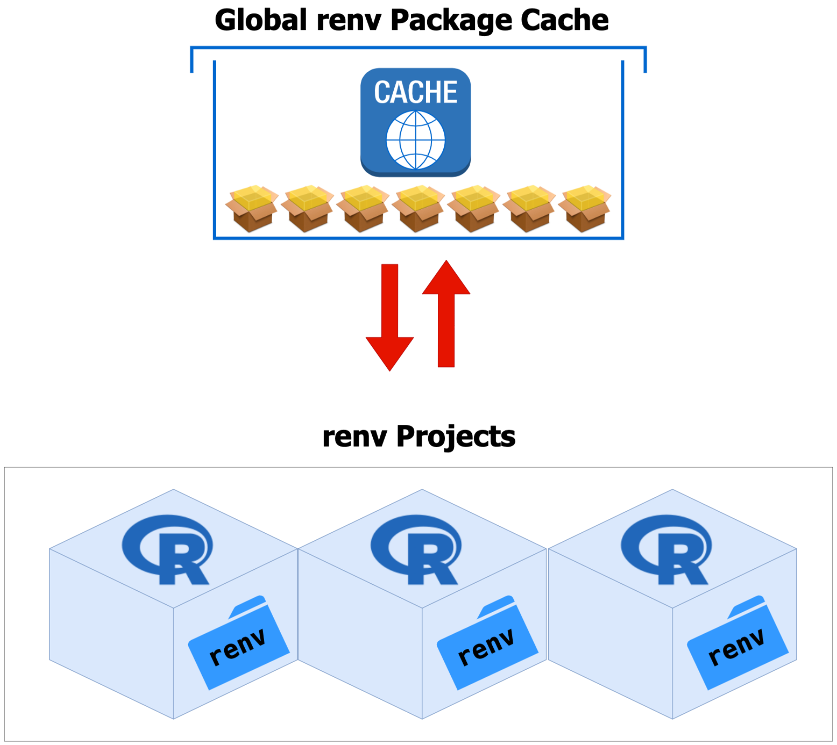


```
{
  "R": {
    "Version": "4.1.0",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cran.rstudio.com"
      }
    ]
  },
  "Packages": {
    "MASS": {
      "Package": "MASS",
      "Version": "7.3-54",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash": "0e59129db205112e3963904db67fd0dc"
    },
    "Matrix": {
      "Package": "Matrix",
      "Version": "1.3-3",
      "Source": "Repository",
      "Repository": "CRAN",
      "Hash": "df57c82e79600601287edfdcef92c2d6"
    },
```

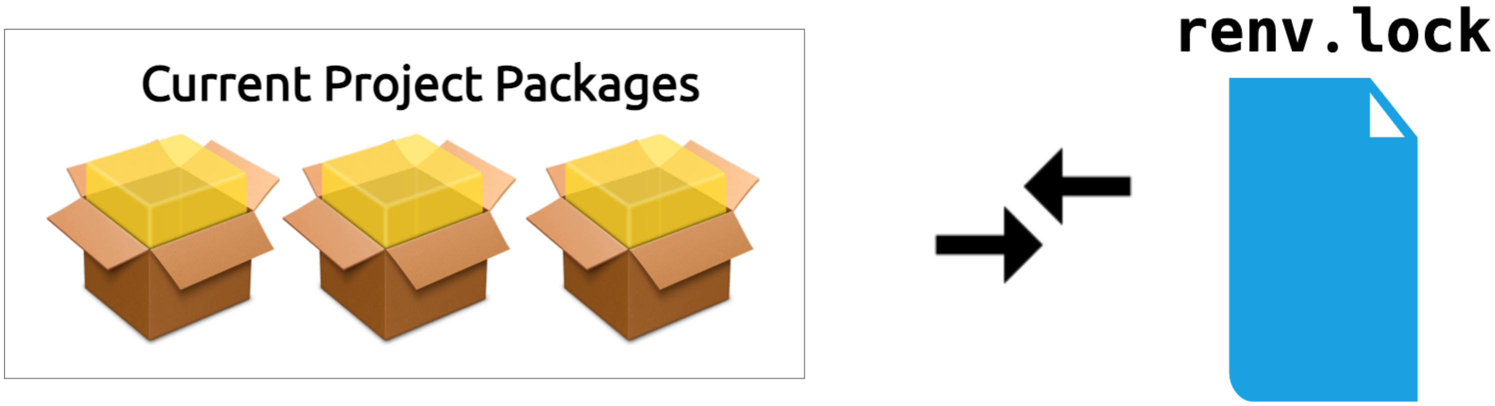
R Version

R Packages

How renv stores packages

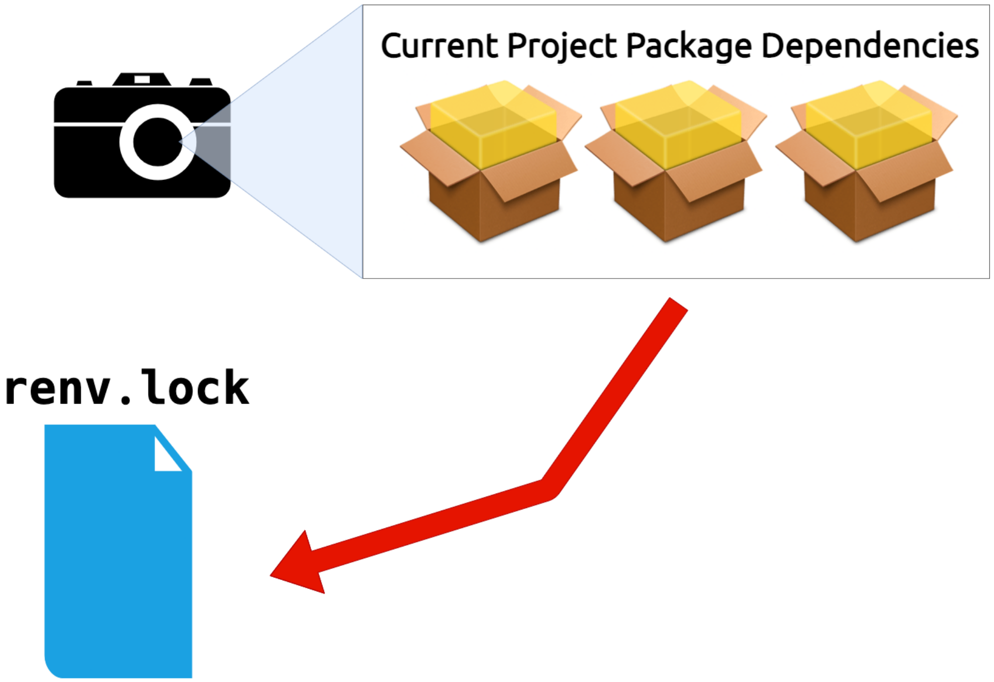


renv::status()



Checks for differences between the renv.lock **and the current project's packages**

renv::snapshot()



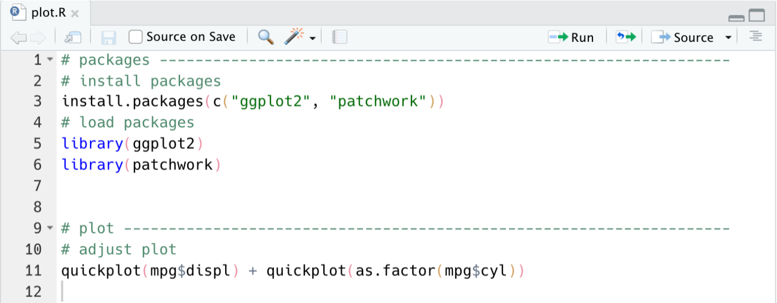
Your Turn 2

In plot.R**, modify your code to load the** patchwork **library, then replace your previous plot with** quickplot(mpg$displ) + quickplot(as.factor(mpg$cyl))

Run renv::status() **in the console**

Run renv::snapshot()**. What's changed?**

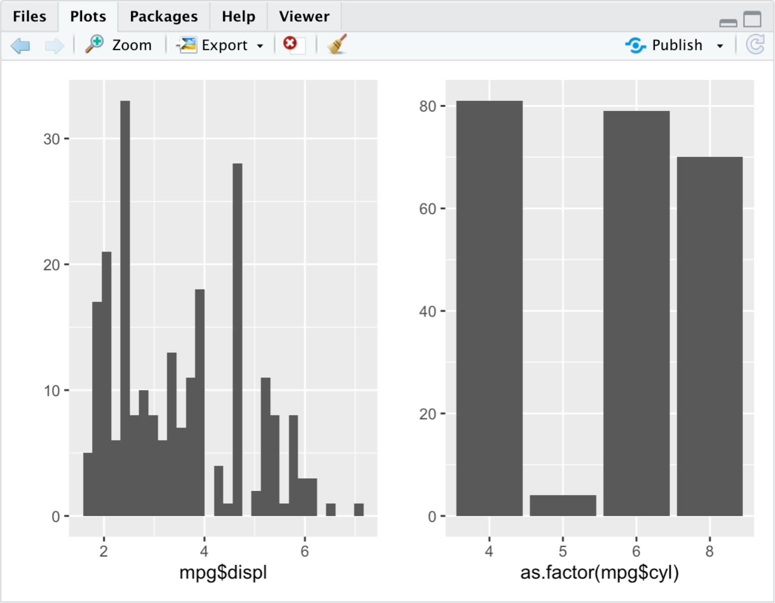
Your Turn 2



```r
# packages ----------------------------------------------------------------
# install packages
install.packages(c("ggplot2", "patchwork"))
# load packages
library(ggplot2)
library(patchwork)


# plot --------------------------------------------------------------------
# adjust plot
quickplot(mpg$displ) + quickplot(as.factor(mpg$cyl))

```



Your Turn 2

```
> renv::status()
The following package(s) are installed but not recorded in the lockfile:

  patchwork  ¯  [1.1.1]

Use `renv::snapshot()` to add these packages to your lockfile.
```

Your Turn 2

```
> renv::snapshot()
The following package(s) will be updated in the lockfile:

# CRAN ===============================
- patchwork   [* -> 1.1.1]

Do you want to proceed? [y/N]:
```

```
Do you want to proceed? [y/N]: y
* Lockfile written to '~/Projects/my-new-project/renv.lock'.
```

renv workflow

1. **Create a project**
2. renv::init()
3. **Write code**
4. renv::snapshot()
5. **Iterate**

Restoring project states

1. **Copy project**
2. renv::restore()
3. **That's it!**

Using renv with targets

targets encourages

`tar_option_set(packages = "...")`

Using renv with targets

targets encourages
tar_option_set(packages = "...")

tar_option_set() **is efficient but unique to targets**

Using renv with targets

```
library(ggplot2)

targets::tar_target()

require(dplyr)

requireNamespace("devtools")
```

Using renv with targets

targets encourages tar_option_set(packages = "...")

tar_option_set() **is efficient but unique to targets**

tar_renv()

Your Turn 3

Create a new project

Load targets and run tar_script() **in the console. Read** _targets.R **and predict what your dependencies are.**

Run renv::dependencies()

Run tar_renv() **and look at** _targets_packages.R **What did targets do here? Why is that important? Run** renv::dependencies() **again and confirm that we can detect all of our dependencies.**

Run renv::init()

Your Turn 3

```
> usethis::create_project("~/Projects/my-newest-project")
✓ Creating '/Users/mjfrigaard/Projects/my-newest-project/'
✓ Setting active project to '/Users/mjfrigaard/Projects/my-newest-project'
✓ Creating 'R/'
✓ Writing 'my-newest-project.Rproj'
✓ Adding '.Rproj.user' to '.gitignore'
✓ Opening '/Users/mjfrigaard/Projects/my-newest-project/' in new RStudio session
✓ Setting active project to '<no active project>'
```

Your Turn 3

tar_script(): _targets.R

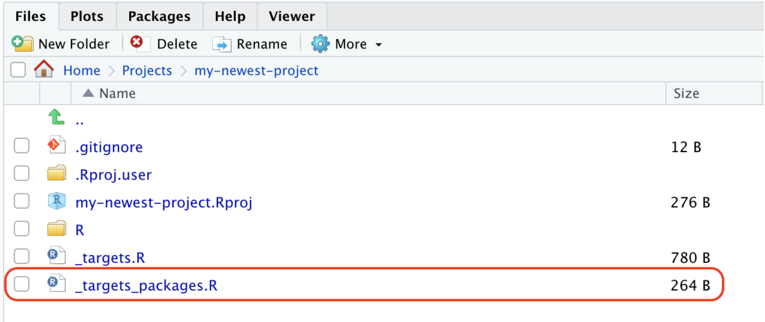
```r
library(targets)
# This is an example _targets.R file. Every {targets} pipeline
# needs one. Use tar_script() to create _targets.R and
# tar_edit() to open it again for editing. Then, run tar_make()
# to run the pipeline and tar_read(summary) to view the results.
# Define custom functions and other global objects. This is where
# you write source(\"R/functions.R\") if you keep your functions
# in external scripts.
summ <- function(dataset) {
  summarize(dataset, mean_x = mean(x))
}
# Set target-specific options such as packages.
tar_option_set(packages = "dplyr")
# End this file with a list of target objects.
list(
  tar_target(data, data.frame(
    x = sample.int(100),
    y = sample.int(100))),
  # Call your custom functions as needed.
  tar_target(summary, summ(data))
)
```

Your Turn 3

```
> renv::dependencies()
Finding R package dependencies ... Done!
                                                   Source Package Require Version   Dev
1 /Users/mjfrigaard/Projects/my-newest-project/_targets.R   dplyr                 FALSE
2 /Users/mjfrigaard/Projects/my-newest-project/_targets.R targets                 FALSE
```

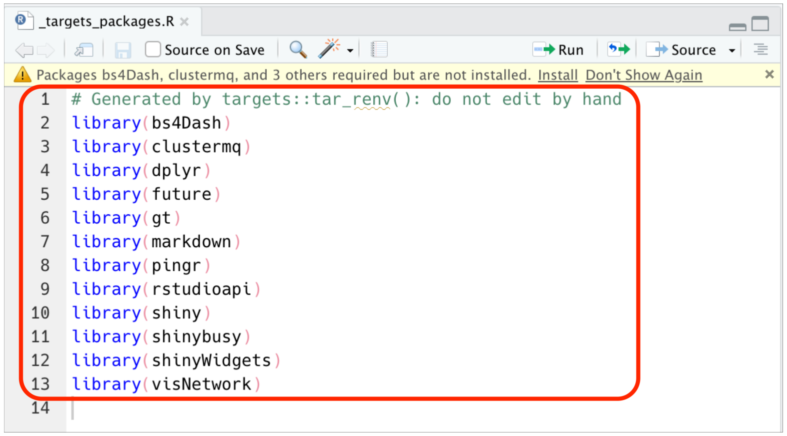
Your Turn 3

tar_renv()



Name	Size
..	
.gitignore	12 B
.Rproj.user	
my-newest-project.Rproj	276 B
R	
_targets.R	780 B
_targets_packages.R	264 B

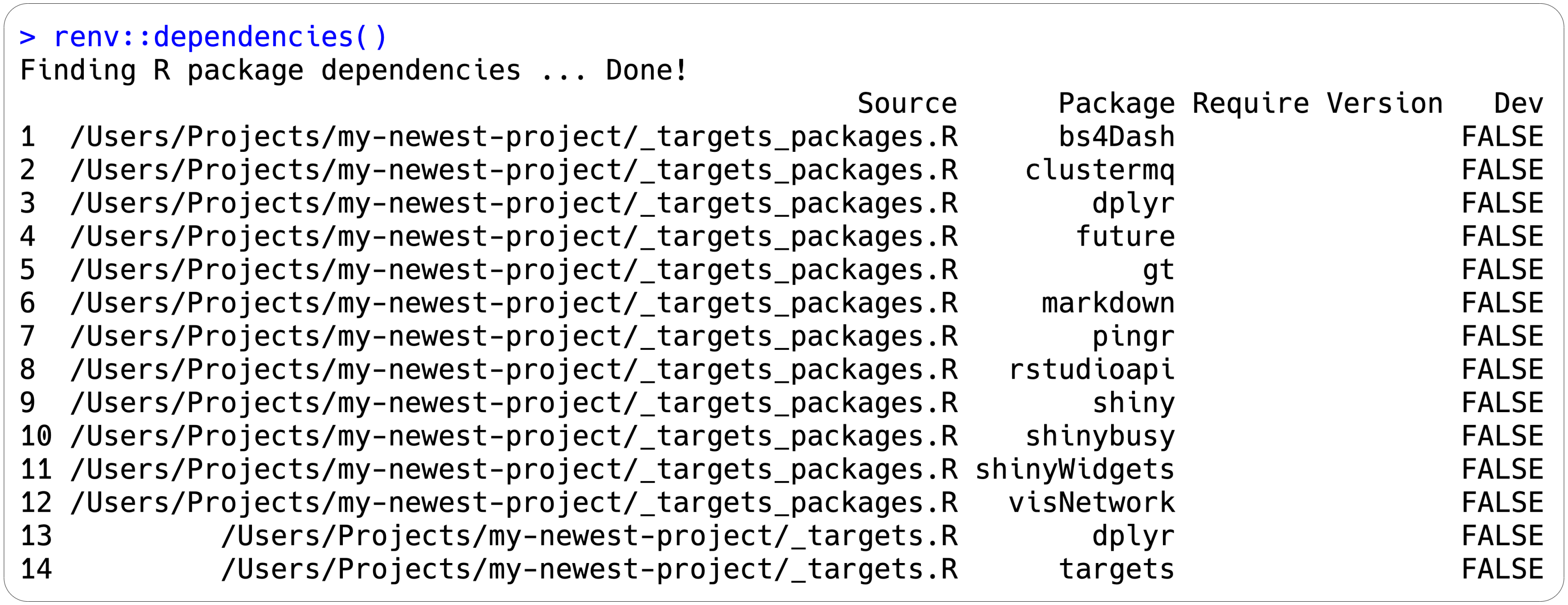
_targets_packages.R



Packages bs4Dash, clustermq, and 3 others required but are not installed.

```r
# Generated by targets::tar_renv(): do not edit by hand
library(bs4Dash)
library(clustermq)
library(dplyr)
library(future)
library(gt)
library(markdown)
library(pingr)
library(rstudioapi)
library(shiny)
library(shinybusy)
library(shinyWidgets)
library(visNetwork)
```

Your Turn 3



```
> renv::dependencies()
Finding R package dependencies ... Done!
                                                  Source      Package Require Version   Dev
1  /Users/Projects/my-newest-project/_targets_packages.R      bs4Dash                 FALSE
2  /Users/Projects/my-newest-project/_targets_packages.R    clustermq                 FALSE
3  /Users/Projects/my-newest-project/_targets_packages.R        dplyr                 FALSE
4  /Users/Projects/my-newest-project/_targets_packages.R       future                 FALSE
5  /Users/Projects/my-newest-project/_targets_packages.R           gt                 FALSE
6  /Users/Projects/my-newest-project/_targets_packages.R     markdown                 FALSE
7  /Users/Projects/my-newest-project/_targets_packages.R        pingr                 FALSE
8  /Users/Projects/my-newest-project/_targets_packages.R   rstudioapi                 FALSE
9  /Users/Projects/my-newest-project/_targets_packages.R        shiny                 FALSE
10 /Users/Projects/my-newest-project/_targets_packages.R    shinybusy                 FALSE
11 /Users/Projects/my-newest-project/_targets_packages.R shinyWidgets                 FALSE
12 /Users/Projects/my-newest-project/_targets_packages.R   visNetwork                 FALSE
13          /Users/Projects/my-newest-project/_targets.R        dplyr                 FALSE
14          /Users/Projects/my-newest-project/_targets.R      targets                 FALSE
```

Your Turn 3

```
> renv::init()
* Initializing project ...
* Discovering package dependencies ... Done!
* Copying packages into the cache ... [73/73] Done!
* Resolving missing dependencies  ...
```

Your Turn 3

```
* Lockfile written to '~/Projects/my-newest-project/renv.lock'.

Restarting R session...

* Project '~/Projects/my-newest-project' loaded. [renv 0.13.2]
```

Resources

renv: Getting started: A brief introduction to renv and its workflows

renv: Project Environments for R (blog): An RStudio blog post introducing renv

renv: Project Environments for R (talk): A talk on renv from rstudio::conf() 2020