R Environment Management

Lingyan Yu

2024-04-20

1. R Environment Management In an R Project

Initiate R environment management by creating a `renv.lock` snapshot and a project-specific library:

```
renv::init()
```

```
## - Linking packages into the project library ... Done!
## - Resolving missing dependencies ...
## # Installing packages --------------------------------------------------------
## The following package(s) will be updated in the lockfile:
##
## # CRAN -----------------------------------------------------------------------
## - base64enc     [* -> 0.1-3]
## - bslib         [* -> 0.7.0]
## - cachem        [* -> 1.0.8]
## - cli           [* -> 3.6.2]
## - digest        [* -> 0.6.35]
## - evaluate      [* -> 0.23]
## - fastmap       [* -> 1.1.1]
## - fontawesome   [* -> 0.5.2]
## - fs            [* -> 1.6.3]
## - glue          [* -> 1.7.0]
## - highr         [* -> 0.10]
## - htmltools     [* -> 0.5.8.1]
## - jquerylib     [* -> 0.1.4]
## - jsonlite      [* -> 1.8.8]
## - knitr         [* -> 1.45]
## - lifecycle     [* -> 1.0.4]
## - memoise       [* -> 2.0.1]
## - mime          [* -> 0.12]
## - R6            [* -> 2.5.1]
## - rappdirs      [* -> 0.3.3]
## - renv          [* -> 1.0.7]
## - rlang         [* -> 1.1.3]
## - rmarkdown     [* -> 2.26]
## - sass          [* -> 0.4.9]
## - tinytex       [* -> 0.50]
## - xfun          [* -> 0.43]
## - yaml          [* -> 2.3.8]
##
## The version of R recorded in the lockfile will be updated:
```

```
## - R              [* -> 4.3.3]
##
## - Lockfile written to "~/Documents/curated_tutorials/R_Project_Workflow/renv.lock".
```

To update the library is to take a new snapshot: `renv::snapshot()`

2. If There Exists Subprojects

Create a `.Rprofile` file in the subproject folder and write `renv::autoload()` into the file.

Opening the subproject session will automatically load the environment configured and maintained in the upper level.

3. Sync the Environment Across Machines

Git only needs to track the `renv.lock` in R projects.

- One may restore the environment by running `renv::restore()` in the R console with the environment snapshot stored in `renv.lock`.
- Actual `renv` library folder is typically too large for GitHub.
- Optional: tracking the `renv` library folder as well to save installation time based on the `renv.lock` snapshot

The restoring process can be automated by adding the following code to the R script:

```r
if (!requireNamespace("renv", quietly = TRUE)) install.packages("renv")
renv::restore()
```

However, the restoring process can be time-consuming and may require some manual resolutions.

One may want to run the script in the background or backup the `renv` folder using cloud storage when switching to a new machine for the first time.