{renv}

Package management for R

What is {renv}?

{renv} is an R package for managing project dependencies and creating reproducible environments

Benefits of using {renv}

1. **Isolation:** Creates project-specific environments separate from the global R library.

2. **Reproducibility:** Ensures consistent package versions for code reproducibility.

3. **Collaboration:** Facilitates sharing and collaborating on projects with others.

Getting Started with {renv}

1. Install {renv}

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

2. Initialize a project

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

3. Install packages

```
1  install.packages("other_package")
2  # only an option when using renv!
3  install.packages("github_user/github_package")
```

4. Track dependencies via a lockfile

```
1  renv::snapshot()
```

Behind the scenes

- Your project `.Rprofile` is updated to include:

```
1  source("renv/activate.R")
```

- This is run every time R starts, and does some management of the library paths to make sure when you call `install.packges("package")` or `library(package)` it does to the right place (`renv/library/R-{version}/{computer-specifics}`)

- A `renv.lock` file (really just a text file) is created to store the names and versions of the packages.

renv.lock

```json
{
  "R": {
    "Version": "4.3.0",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cran.rstudio.com"
      }
    ]
  },
  "Packages": {
    "R6": {
      "Package": "R6",
      "Version": "2.5.1",
      "Source": "Repository",
      "Repository": "CRAN",
      "Requirements": [
        "R"
      ],
      "Hash": "470851b6d5d0ac559e9d01bb352b4021"
    },
    "base64enc": {
      "Package": "base64enc",
      "Version": "0.1-3",
      "Source": "Repository",
      "Repository": "CRAN",
      "Requirements": [
```

```
        "R"
    ],
    "Hash": "543776ae6848fde2f48ff3816d0628bc"
},
```

Using {renv} later

Restore an environment

```
1  renv::restore()
```

Install new packages

```
1  install.packages("other_package")
```

Update the lockfile

```
1  renv::snapshot()
```

Collaboration with {renv}

- Share the project's `renv.lock` file with collaborators to ensure consistent environments

- When they run `renv::restore()`, the correct versions of the packages will be installed on their computer

```
1  renv::restore()
```

Other helpful functions

Remove packages that are no longer used:

```
1  renv::clean()
```

Check the status of the project library with respect to the lockfile:

```
1  renv::status()
```

This will tell you to `renv::snapshot()` to add packages you've installed but haven't snapshotted, or `renv::restore()` if you're missing packages you need but which aren't installed

Conclusion

`{renv}` benefits:

- Isolation, reproducibility, and collaboration

Getting started with `{renv}`

1. Initialize a project using `renv::init()`
2. Install packages and store with `renv::snapshot()`
3. Restore later or elsewhere with `renv::restore()`

Exercises

3. Install a new R package of your choice. (Not sure what to choose? Try one of these fun packages. For example, I did `install.packages("hadley/emo")`.)

4. Create an R script and save it in your R project. Include some code that requires the package. For example:

```
1  emo::ji("banana")
```

4. Run `renv::status()` to make sure your project picked up the package as a dependency.

5. Run `renv::snapshot()` to record that package in your lockfile.

6. Open your lockfile and look for your new package. For example, mine now has:

```
"emo": {
    "Package": "emo",
    "Version": "0.0.0.9000",
    "Source": "git",
    "RemoteType": "git",
    "RemoteUrl": "https://github.com/hadley/emo",
    "RemoteHost": "api.github.com",
    "RemoteUsername": "hadley",
    "RemoteRepo": "emo",
    "RemoteRef": "master",
    "RemoteSha": "3f03b11491ce3d6fc5601e210927eff73bf8e350",
    "Requirements": [
      "R",
      "assertthat",
      "crayon",
      "glue",
      "lubridate",
      "magrittr",
```