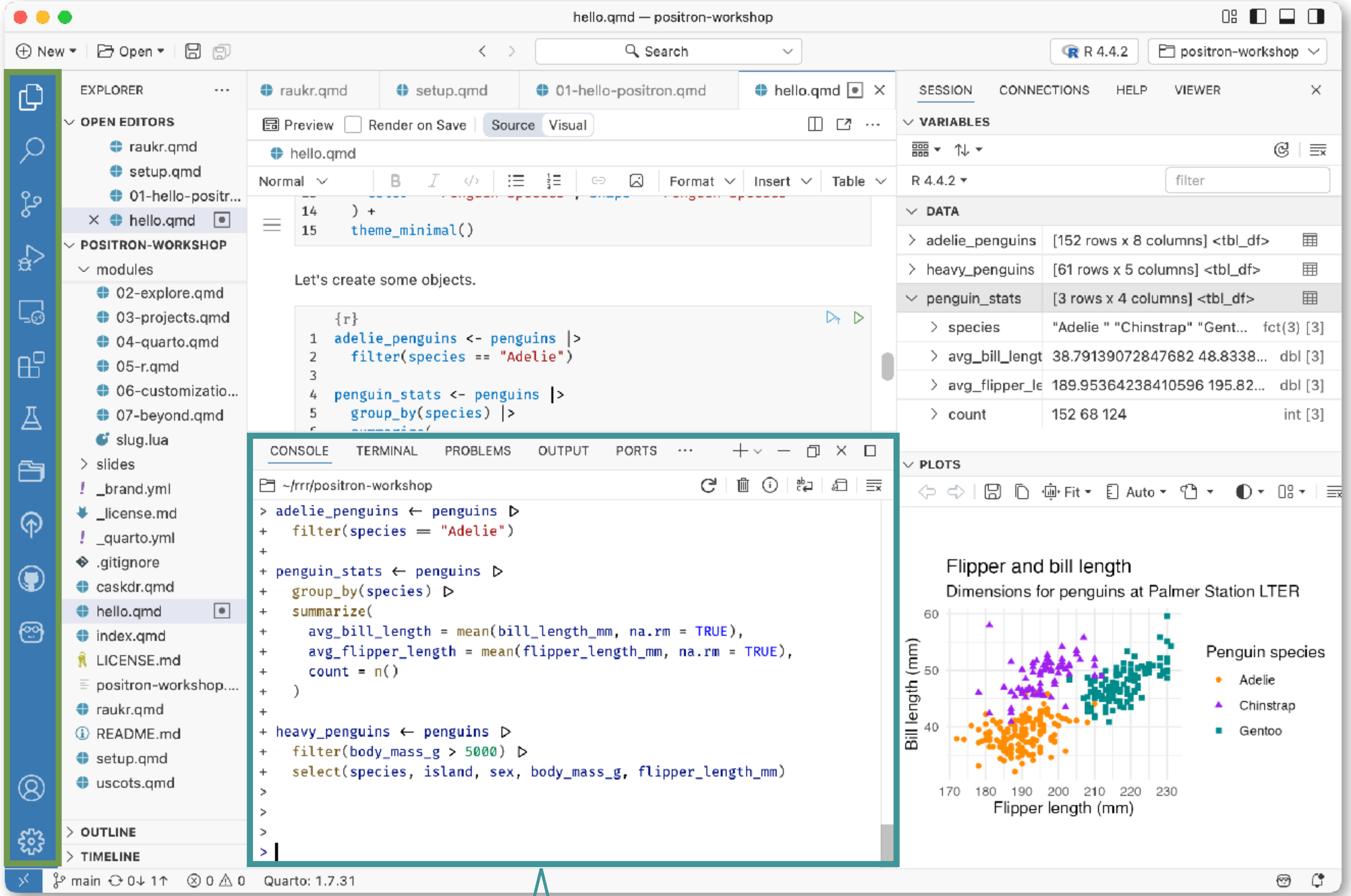
R Programming

Part of "Doing data science in Positron" workshop

Posit Software, PBC



```
  ) +
  theme_minimal()
```

Let's create some objects.

```
{r}
adelie_penguins <- penguins |>
  filter(species == "Adelie")

penguin_stats <- penguins |>
  group_by(species) |>
```

```
> adelie_penguins <- penguins |>
+   filter(species == "Adelie")
+
+ penguin_stats <- penguins |>
+   group_by(species) |>
+   summarize(
+     avg_bill_length = mean(bill_length_mm, na.rm = TRUE),
+     avg_flipper_length = mean(flipper_length_mm, na.rm = TRUE),
+     count = n()
+   )
+
+ heavy_penguins <- penguins |>
+   filter(body_mass_g > 5000) |>
+   select(species, island, sex, body_mass_g, flipper_length_mm)
>
>
>
>
```

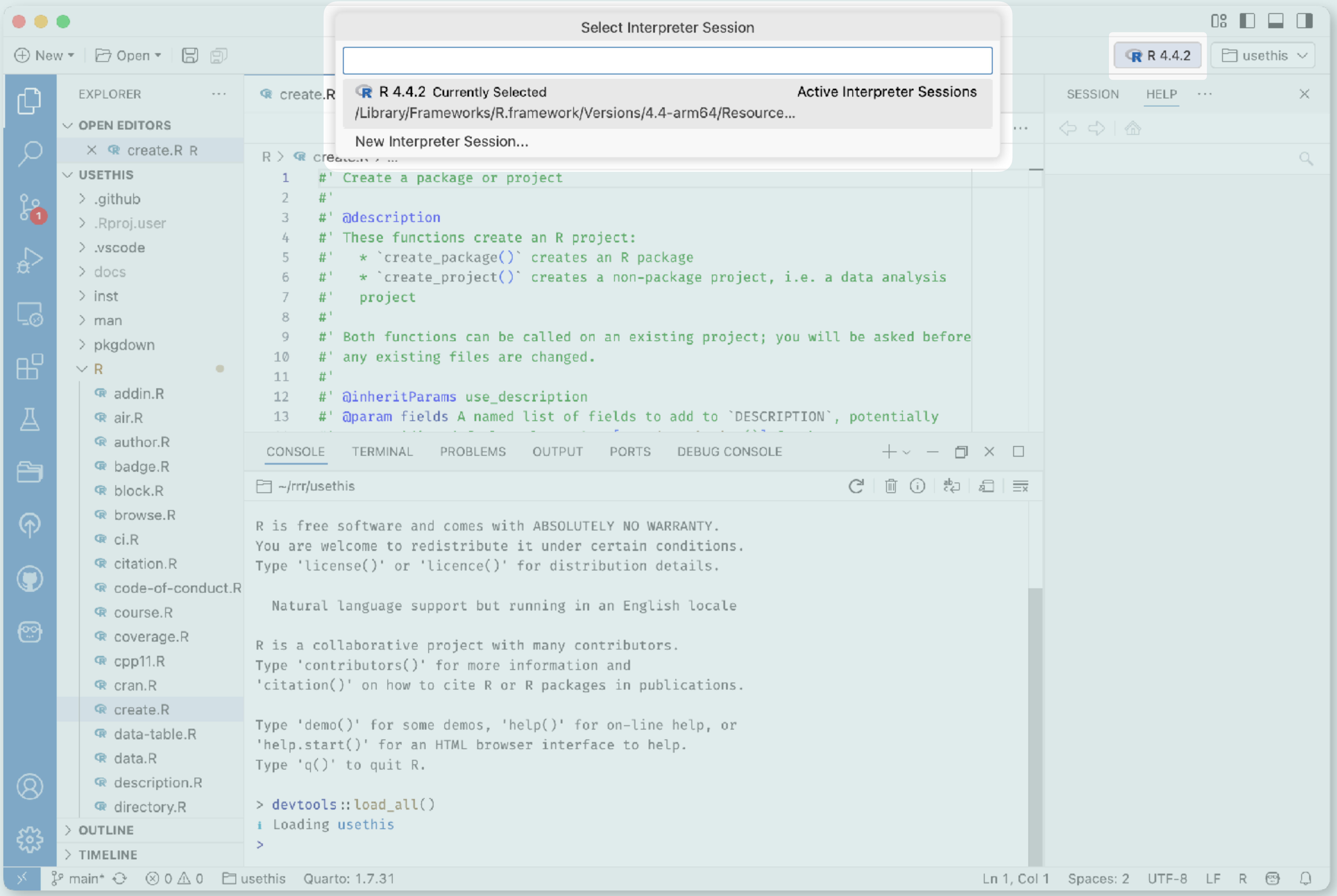
adelie_penguins	[152 rows x 8 columns] <tbl_df>	
heavy_penguins	[61 rows x 5 columns] <tbl_df>	
penguin_stats	[3 rows x 4 columns] <tbl_df>	
species	"Adelie " "Chinstrap" "Gent...	fct(3) [3]
avg_bill_lengt	38.79139072847682 48.8338...	dbl [3]
avg_flipper_le	189.95364238410596 195.82...	dbl [3]
count	152 68 124	int [3]

An R session in the Console

Multiple versions of R

R interpreter sessions

- Positron readily discovers and offers multiple versions of R
- Positron can have multiple, concurrent interpreter sessions, that can be
  - a mix of different R versions
  - a mix of R and Python sessions
  - multiple instances of a single R version



Select Interpreter Session

R 4.4.2 Currently Selected — Active Interpreter Sessions

/Library/Frameworks/R.framework/Versions/4.4-arm64/Resource...

New Interpreter Session...

```r
#' Create a package or project
#'
#' @description
#' These functions create an R project:
#'   * `create_package()` creates an R package
#'   * `create_project()` creates a non-package project, i.e. a data analysis
#'   project
#'
#' Both functions can be called on an existing project; you will be asked before
#' any existing files are changed.
#'
#' @inheritParams use_description
#' @param fields A named list of fields to add to `DESCRIPTION`, potentially
```

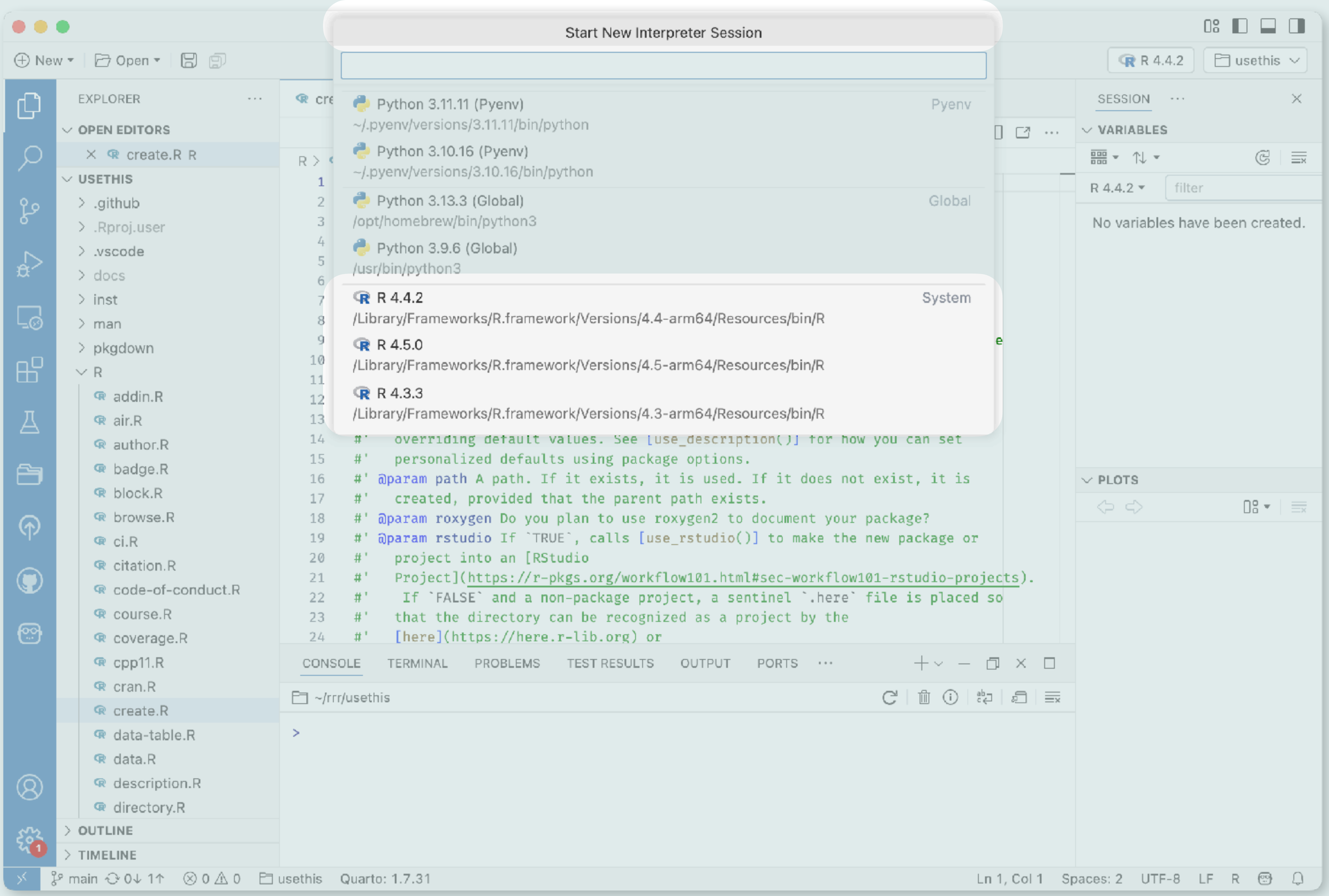
```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

> devtools::load_all()
ℹ Loading usethis
>
```



Start New Interpreter Session

Python 3.11.11 (Pyenv) — Pyenv
~/.pyenv/versions/3.11.11/bin/python

Python 3.10.16 (Pyenv)
~/.pyenv/versions/3.10.16/bin/python

Python 3.13.3 (Global) — Global
/opt/homebrew/bin/python3

Python 3.9.6 (Global)
/usr/bin/python3

R 4.4.2 — System
/Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/bin/R

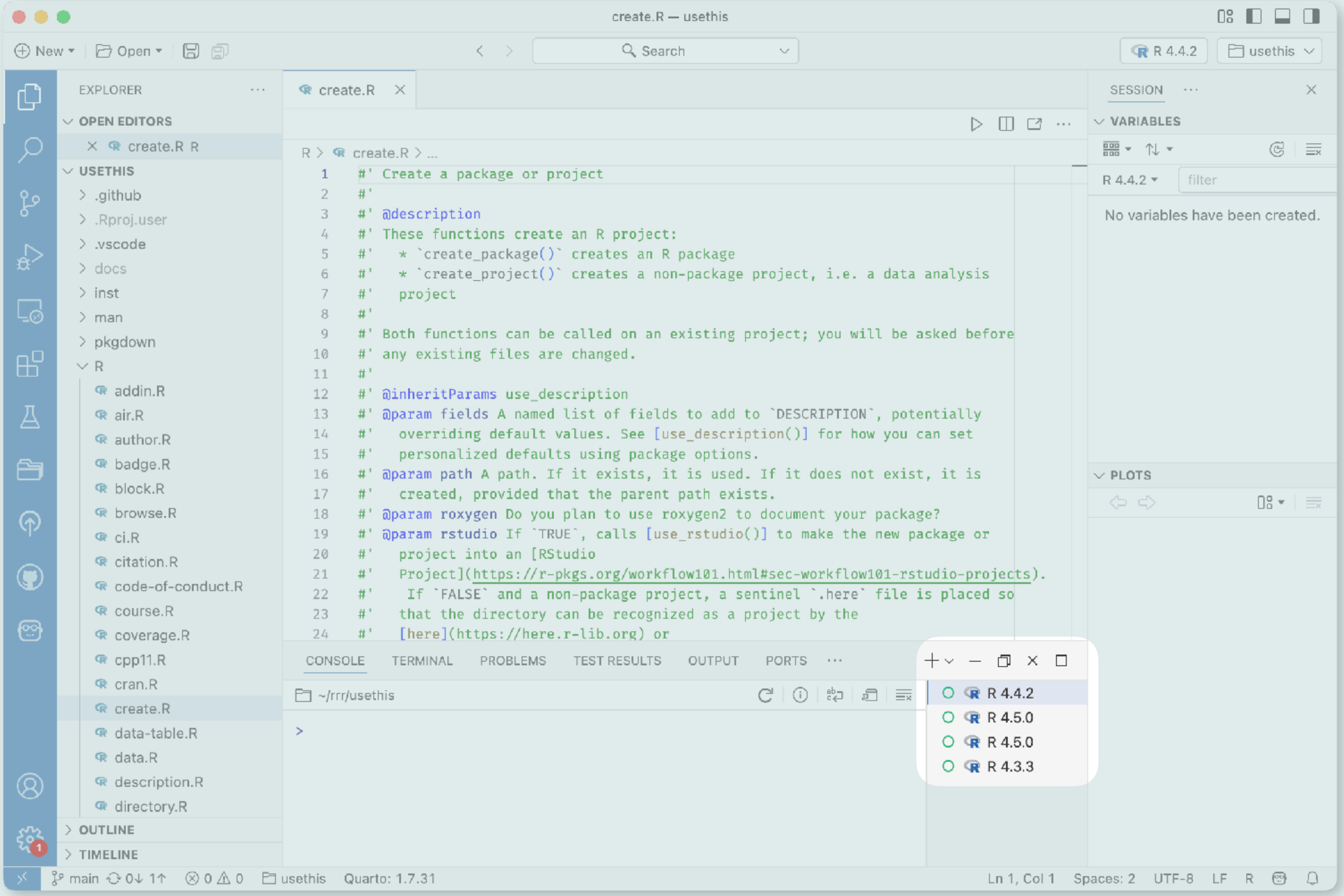
R 4.5.0
/Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/bin/R

R 4.3.3
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/bin/R

```
#'   overriding default values. See [use_description()] for how you can set
#'   personalized defaults using package options.
#' @param path A path. If it exists, it is used. If it does not exist, it is
#'   created, provided that the parent path exists.
#' @param roxygen Do you plan to use roxygen2 to document your package?
#' @param rstudio If `TRUE`, calls [use_rstudio()] to make the new package or
#'   project into an [RStudio
#'   Project](https://r-pkgs.org/workflow101.html#sec-workflow101-rstudio-projects).
#'    If `FALSE` and a non-package project, a sentinel `.here` file is placed so
#'   that the directory can be recognized as a project by the
#'   [here](https://here.r-lib.org) or
```

No variables have been created.

```
~/rrr/usethis
>
```



```r
#' Create a package or project
#'
#' @description
#' These functions create an R project:
#'   * `create_package()` creates an R package
#'   * `create_project()` creates a non-package project, i.e. a data analysis
#'   project
#'
#' Both functions can be called on an existing project; you will be asked before
#' any existing files are changed.
#'
#' @inheritParams use_description
#' @param fields A named list of fields to add to `DESCRIPTION`, potentially
#'   overriding default values. See [use_description()] for how you can set
#'   personalized defaults using package options.
#' @param path A path. If it exists, it is used. If it does not exist, it is
#'   created, provided that the parent path exists.
#' @param roxygen Do you plan to use roxygen2 to document your package?
#' @param rstudio If `TRUE`, calls [use_rstudio()] to make the new package or
#'   project into an [RStudio
#'   Project](https://r-pkgs.org/workflow101.html#sec-workflow101-rstudio-projects).
#'    If `FALSE` and a non-package project, a sentinel `.here` file is placed so
#'   that the directory can be recognized as a project by the
#'   [here](https://here.r-lib.org) or
```

```
>
```

💡 Teaching tip

- It’s best for all students in your course to be using the same R version.
- But sometimes you might not want to update (or, more likely, downgrade) your “personal” R version to match the course version.
- Testing out your course materials with the version of R your students are using without changing your default R version can be helpful.

Air

Air formatter for R

posit-dev.github.io/air

- When you install Positron, you get Air "for free” as a pre-bundled extension.
- Air formats your R code based on some opiniated but also widely-used and agreed-upon conventions — details at posit-dev.github.io/air/formatter.html.

Air + Positron: practical suggestions

- Air extension ships with Positron and includes the Air binary so Air should just work in Positron.
- You can configure Air to be used at the
  - user level to say “I use Air to format the R code in all of my projects”: by adjusting your user setting (more on this in the next module) or
  - workspace level to say “We use Air to format the R code in this project”: by adjusting your workspace settings (more on this in the next module too) or with `usethis::use_air()`*
- You can also use Air “one-off” from the *Command palette:*
  - *Air: Format Workspace Folder*
  - *Air: Format Document*
  - *Air: Format Selection*
- The Git diff is a great way to see what has changed, so inspect it before committing and pushing!
- There are various ways to disable Air formatting of a specific file, line (or lines in a pipeline), etc.

* Currently requires dev version of usethis

3m 00s

Your turn

Format some ugly R code

- The example project we downloaded earlier has a file with poorly formatted R code: `air-practice.R`
- Open it in Positron. Alternatively, open a personal R file with questionable formatting.
- Remove the `# fmt: skip file` line, so that Air will format the file.
- Experiment with the *Format Selection* and/or *Format Document* commands to see how Air would reformat it.
- Inspect the Git diff to see what's changed.

If you ❤️ Air *(and you will ❤️ Air)*

Turn on Format on Save for R documents by opening your `settings.json`* and adding:

```
{
    "[r]": {
        "editor.formatOnSave": true,
        "editor.defaultFormatter": "Posit.air-vscode"
    }
}
```

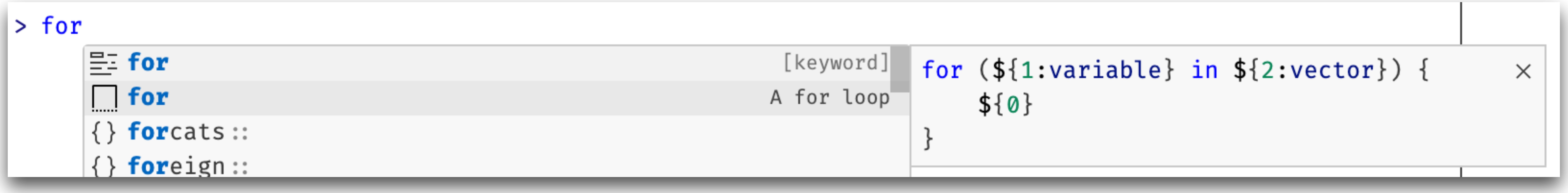
* More on settings in the next module!

Snippets

Snippets

- Positron's R support provides a few snippets related to R's reserved words.
- Positron provides fewer built-in snippets than RStudio.
- You can configure additional snippets at the user or workspace level.
- Positron uses TextMate syntax for snippets, inherited from VS Code. This is different from RStudio's snippet syntax.
- Snippets are typically inserted via the usual completions offered by IntelliSense. There's also a dedicated command: *Insert Snippet*.
- https://positron.posit.co/r-snippets.html

Built-in snippet example: for loop



```
> for
for        [keyword]
for        A for loop
{} forcats ::
{} foreign ::
```

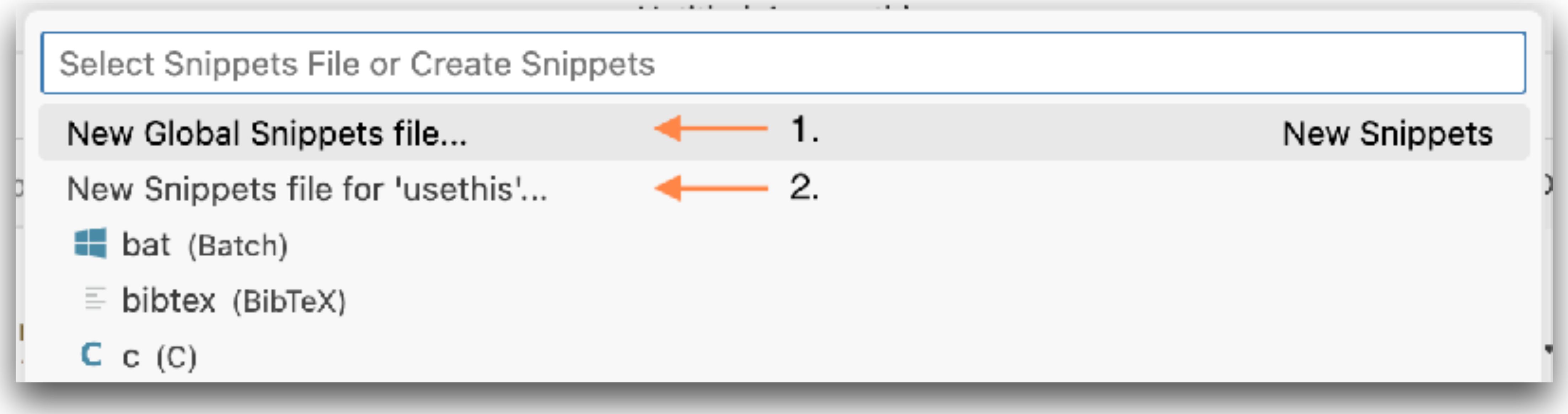
```
for (${1:variable} in ${2:vector}) {
    ${0}
}
```

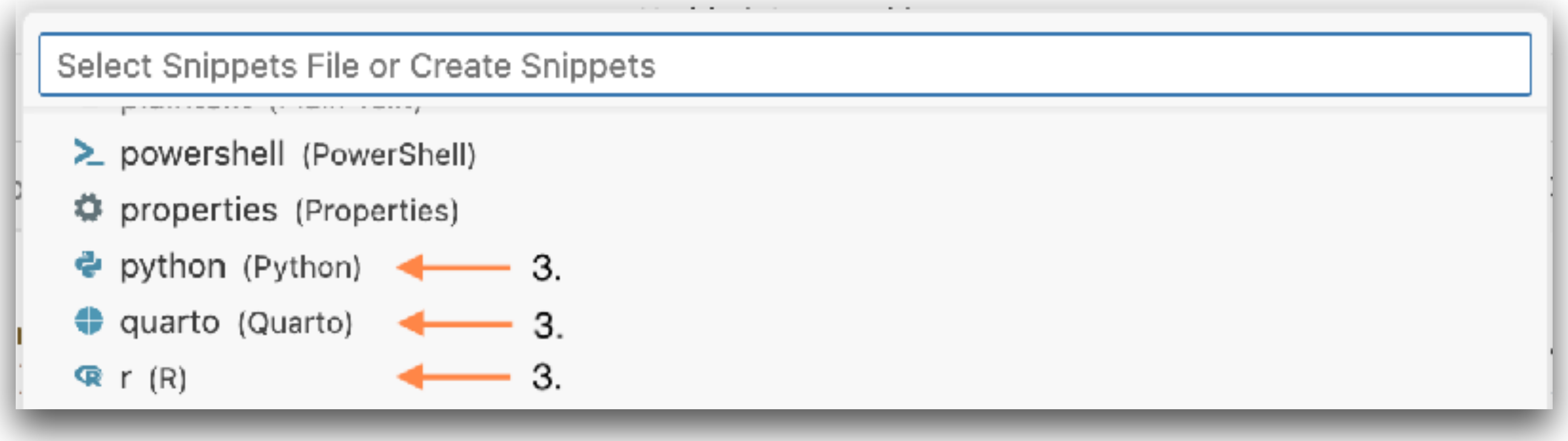
👆helps you create code like 👇

```
for (variable in vector) {
    # code to repeat
}
```

How to configure your own snippets

Command palette: *Snippets: Configure Snippets*


Select Snippets File or Create Snippets
New Global Snippets file...
1.
New Snippets
New Snippets file for 'usethis'...
2.
bat (Batch)
bibtex (BibTeX)
c (C)



https://positron.posit.co/r-snippets.html

1. Global Snippets file: User-level. Potentially more than 1 language.
2. Workspace-specific file: Specific to 1 workspace. Potentially more than 1 language.
3. Language-specific file: User-level.