

Customization + extensibility

Part of "Doing data science in Positron" workshop



Settings

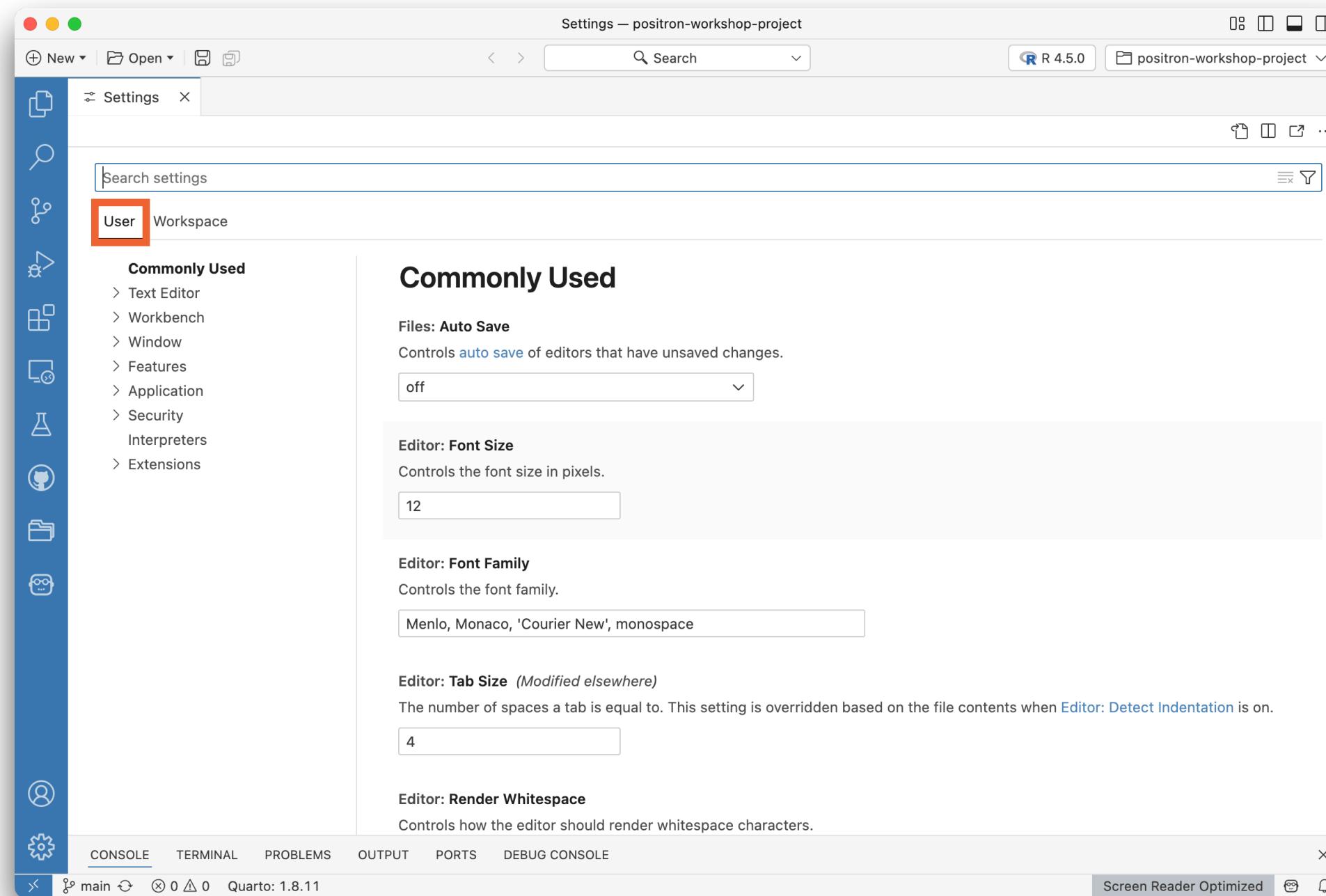
Settings

- Nearly every part of Positron's editor, user interface, and functional behavior has customizable options.
- Settings can be customized at two different scopes:
 - **User settings:** Settings that apply globally to any instance of Positron you open.
 - **Workspace settings:** Settings stored inside your workspace (folder) and only apply when the workspace is opened.
- Positron stores setting values in a settings JSON file.
- Settings can be changed by:
 - editing the settings JSON file or
 - using the Settings editor interface

Updating user settings

To update using the UI:

- In the Command Palette:
Preferences: Open User Settings
- In the top menu:
Positron > Settings... > Settings



To edit the JSON file:

- In the Command Palette:
Preferences: Open User Settings (JSON)

A screenshot of a code editor showing the file "settings.json" open. The status bar at the bottom indicates the full path: "Users > mine > Library > Application Support > Positron > User > settings.json". A red box highlights this path. The code editor shows the following JSON content:

```
// Settings imported from Visual Studio Code
{
  "quarto.visualEditor.markdownWrap": "sentence",
  "positron.assistant.enable": true,
  "positron.assistant.enabledProviders": [
    "anthropic"
  ],
  "git.confirmSync": true
}
```

The code editor has a dark theme with light-colored syntax highlighting. The status bar also shows other details like "R 4.5.0", "positron-workshop-project", "CONSOLE", "TERMINAL", "PROBLEMS", "OUTPUT", "PORTS", "DEBUG CONSOLE", and file navigation information.

Path to
settings.json

3m 00s

Your turn

Customize user settings

- Update your user settings to change the Icon Theme to Minimal via the Settings editor interface.
 - Hint: Filter for settings with the word “icon”.
- Open the user `settings.json` file to confirm the change you made.
- In the user `settings.json` file, change the Icon Theme to `vs-seti` and save your changes.
- Go back to the Icon Theme settings in the Settings editor interface and confirm the change you made in the JSON file.

Settings — positron-workshop-project

New | Open | Search R 4.5.0 positron-workshop-project

Settings X

icon 16 Settings Found

User Workspace

Text Editor (2)
Diff Editor (1)
Suggestions (1)

Workbench (7)
Appearance (2)
Breadcrumbs (1)
Editor Management (1)
Window (1)

Features (5)
Explorer (1)
Testing (1)
Source Control (1)
Terminal (2)

Workbench > Activity Bar: Icon Click Behavior
Controls the behavior of clicking an Activity Bar icon in the workbench. This value is ignored when [Workbench > Activity Bar: Location](#) is not set to default.

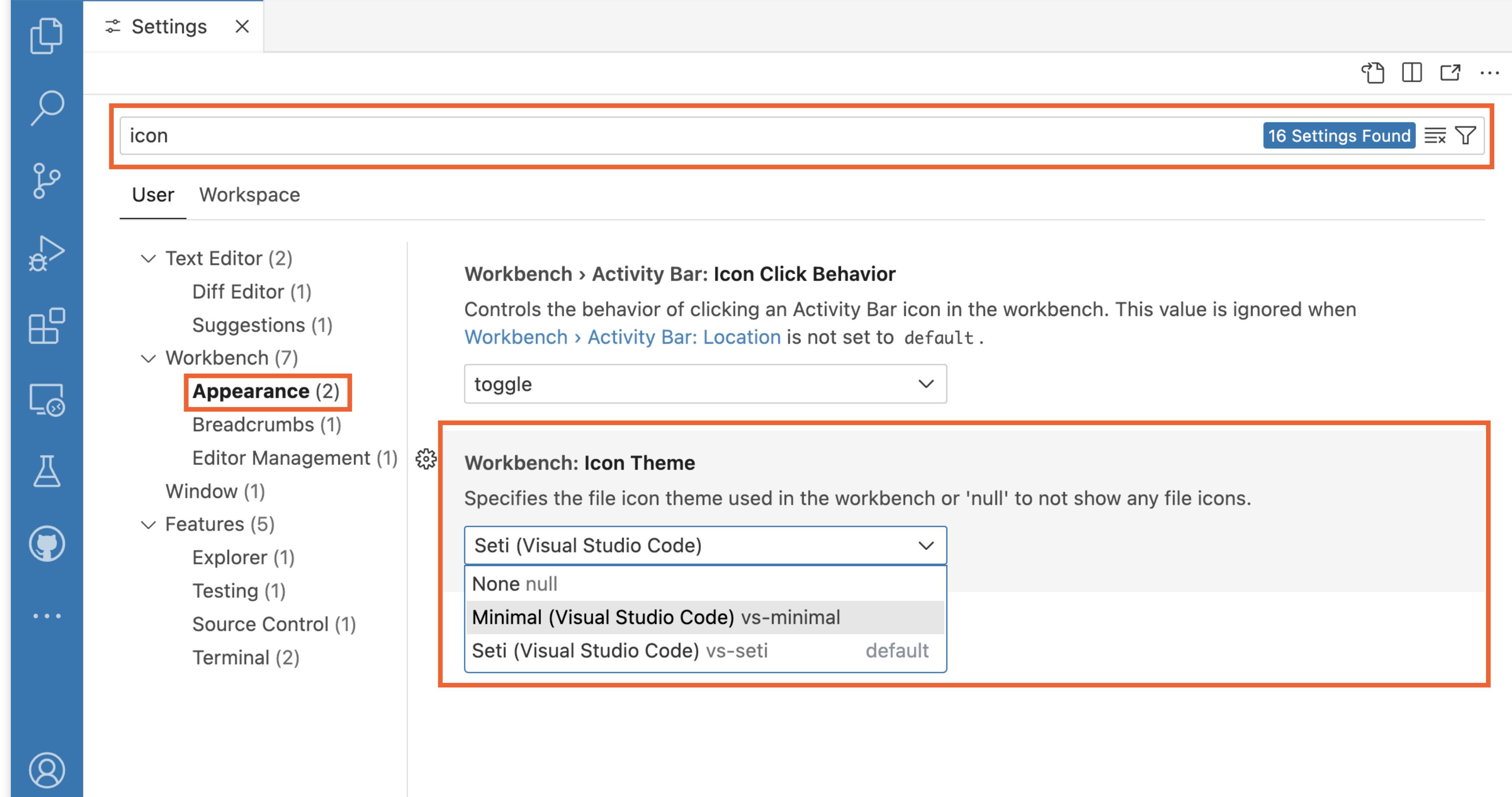
toggle

Workbench: Icon Theme
Specifies the file icon theme used in the workbench or 'null' to not show any file icons.

Seti (Visual Studio Code)
None null
Minimal (Visual Studio Code) vs-minimal
Seti (Visual Studio Code) vs-seti default

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE

main 0 0 Quarto: 1.8.11 Screen Reader Optimized



settings.json — positron-workshop-project

⊕ New | ⌂ Open | ⌂ ⌂ | < > | Search | R 4.5.0 | positron-workshop-project

Settings settings.json

Users > mine > Library > Application Support > Positron > User > settings.json > ...

```
1 // Settings imported from Visual Studio Code
2 {
3     "quarto.visualEditor.markdownWrap": "sentence",
4     "positron.assistant.enable": true,
5     "positron.assistant.enabledProviders": [
6         "anthropic"
7     ],
8     "git.confirmSync": true,
9     "workbench.iconTheme": "vs-minimal"
10 }
```

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE

main /opt/homebrew/bin/python3 0 0 Quarto: 1.8.11 Ln 1, Col 1 Tab Size: 4 UTF-8 LF {} JSON with Comments

A screenshot of the Visual Studio Code interface showing the `settings.json` file for the `positron-workshop-project`. The file path is `Users > mine > Library > Application Support > Positron > User > settings.json`. The code editor displays the following JSON configuration:

```
1 // Settings imported from Visual Studio Code
2 {
3     "quarto.visualEditor.markdownWrap": "sentence",
4     "positron.assistant.enable": true,
5     "positron.assistant.enabledProviders": [
6         "anthropic"
7     ],
8     "git.confirmSync": true,
9     "workbench.iconTheme": "vs-minimal"
10 }
```

The line `"workbench.iconTheme": "vs-minimal"` is highlighted with a red rectangle. A callout bubble with a black border and arrow points to this line, containing the text: **Change to “vs-seti”**.

The VS Code interface includes a sidebar with icons for files, search, and other tools. The bottom navigation bar shows tabs for CONSOLE, TERMINAL, PROBLEMS, OUTPUT, PORTS, and DEBUG CONSOLE. The status bar at the bottom provides information about the current file (`main`), terminal (`/opt/homebrew/bin/python3`), problems (`0` errors, `0` warnings), Quarto version (`1.8.11`), and file details (`Ln 1, Col 1`, `Tab Size: 4`, `UTF-8 LF`, `{ } JSON with Comments`).

The screenshot shows the RStudio IDE interface for a Quarto project named "positron-workshop-project". The title bar indicates "hello.qmd – positron-workshop-project". The top menu bar includes "SESSION", "HELP", "VIEWER", and "CONNECTIONS". The left sidebar features the "EXPLORER" tab, which is expanded to show the project structure under "POSITRON-WORKSHOP-PROJECT". The "OPEN EDITORS" tab is also visible. The main workspace displays the content of the "hello.qmd" file, which contains R code for generating a Quarto document. The code includes a title, a plot-penguins section, and a library loading section. The bottom status bar shows "Charlotte Wickham (3 weeks ago)" and "Quarto: 1.8.14".

```
1 ---  
2 title: "Hello, Quarto"  
3 html: default  
4 pdf: default  
5 ---  
6 ---  
7 >Run Cell | Run Next Cell  
8 ``{r}  
9 #| label: load-packages  
10 #| include: false  
11  
12 library(tidyverse)  
13 library(palmerpenguins)  
14 ---  
15  
16 ## Meet Quarto  
17
```

This screenshot is similar to the first one, showing the RStudio IDE with the same project and file content. However, a tooltip has appeared over the "Hello.qmd" file in the "OPEN EDITORS" list, displaying the full path: "~/Documents/positron/positron-workshop-project/fruit_avg_browser.R". The rest of the interface and code content are identical to the first screenshot.

```
1 ---  
2 title: "Hello, Quarto"  
3 format:  
4 html: default  
5 pdf: default  
6 ---  
7 >Run Cell | Run Next Cell  
8 ``{r}  
9 #| label: load-packages  
10 #| include: false  
11  
12 library(tidyverse)  
13 library(palmerpenguins)  
14 ---  
15  
16 ## Meet Quarto  
17
```

Workspace settings

- Workspace settings are specific to a project and override user settings.
- Positron (like VS Code) stores workspace settings at the root of the project in a `.vscode` folder, making it easy to share settings with others in a version-controlled (e.g., Git) project.

Updating workspace settings

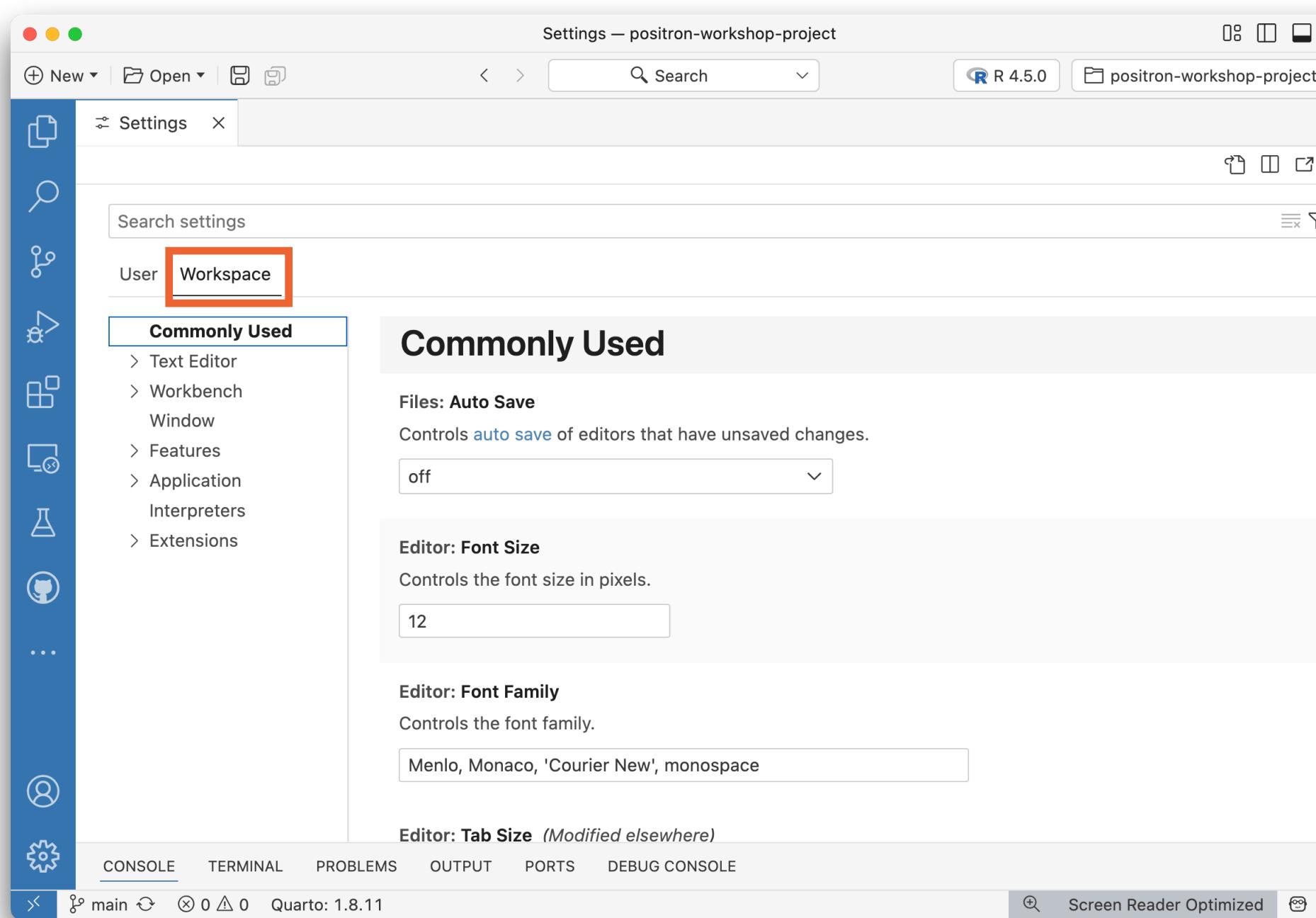
To update using the UI:

- In the Command Palette:

Preferences: Open Workspace Settings

- In the top menu:

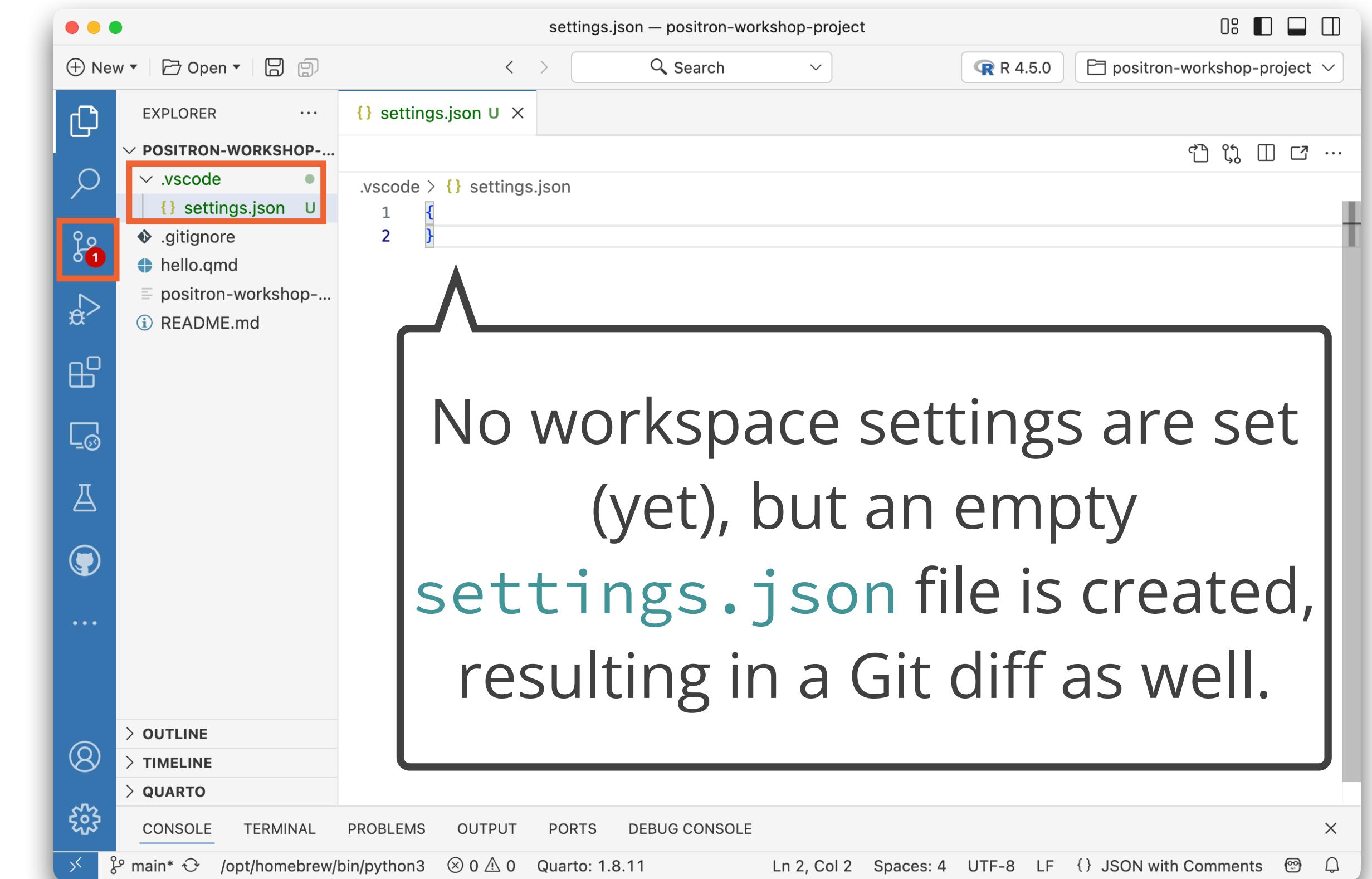
Positron > Settings... > Settings



To edit the JSON file:

In the Command Palette:

Preferences: Open Workspace Settings (JSON)

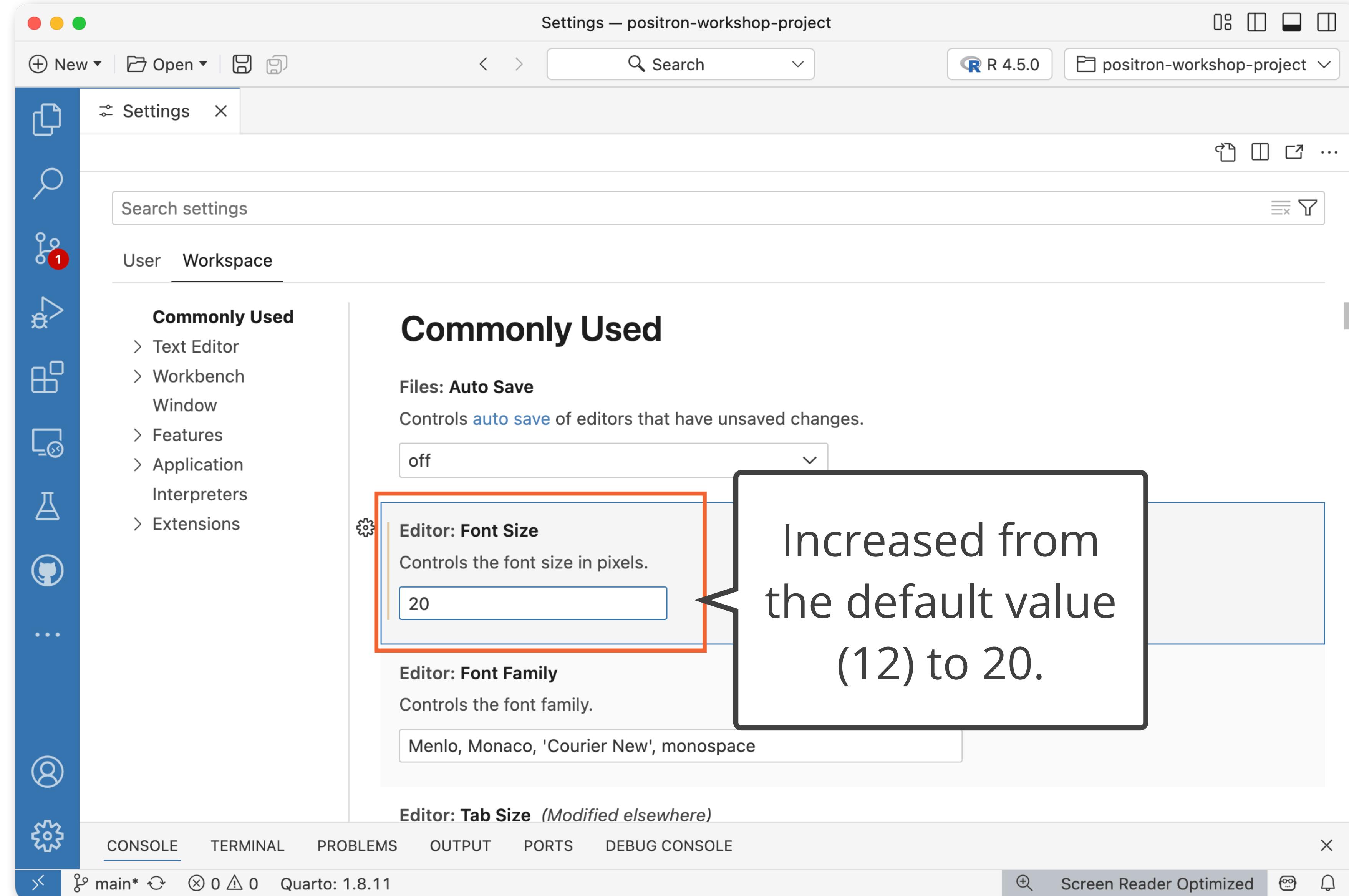


3m 00s

Your turn

Customize workspace settings

- Update your workspace settings in the `positron-workshop-project` using the Settings editor interface.
 - You can update whatever setting you like.
 - If you need inspiration, change the font size in the editor.
- Open the `settings.json` file in the `.vscode` folder of your project to confirm the change you made.
- Click on the Source Control icon in the Activity bar and view the diff created by updating your settings. Contrast this to what happened (or didn't happen) in the Source Control when you updated your user settings.

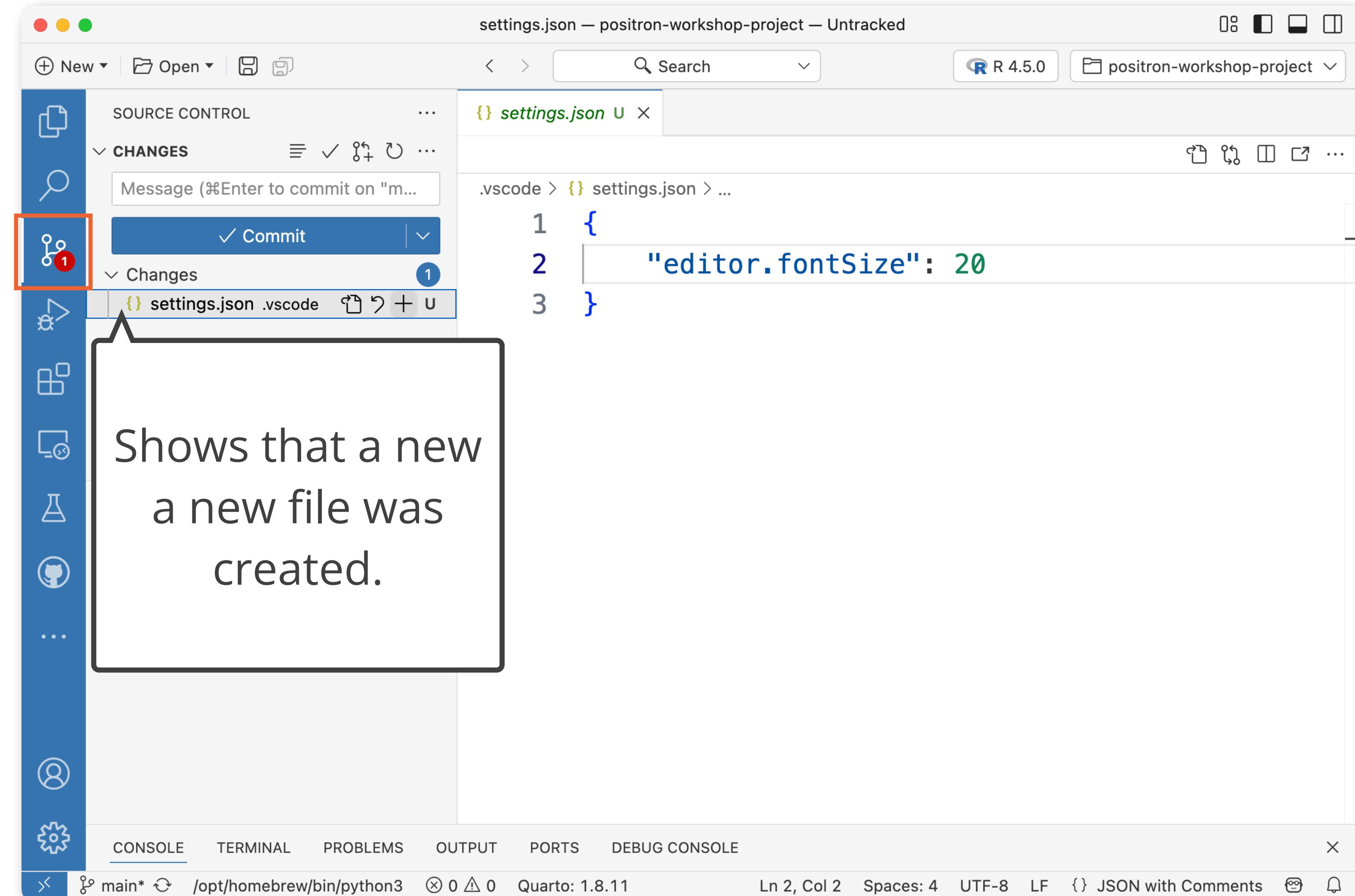


The screenshot shows the VS Code interface with the following details:

- Title Bar:** settings.json — positron-workshop-project — Untracked
- Toolbar:** New, Open, Settings, Search, R 4.5.0, positron-workshop-project
- Left Sidebar (Explorer):** Shows the project structure:
 - POSIX-style folder icons
 - .vscode folder
 - settings.json (Untracked)
 - .gitignore
 - hello.qmd
 - positron-workshop-...
 - README.md
- Code Editor:** The file settings.json is open, showing the following JSON code:

```
1 {  
2     "editor.fontSize": 20  
3 }
```

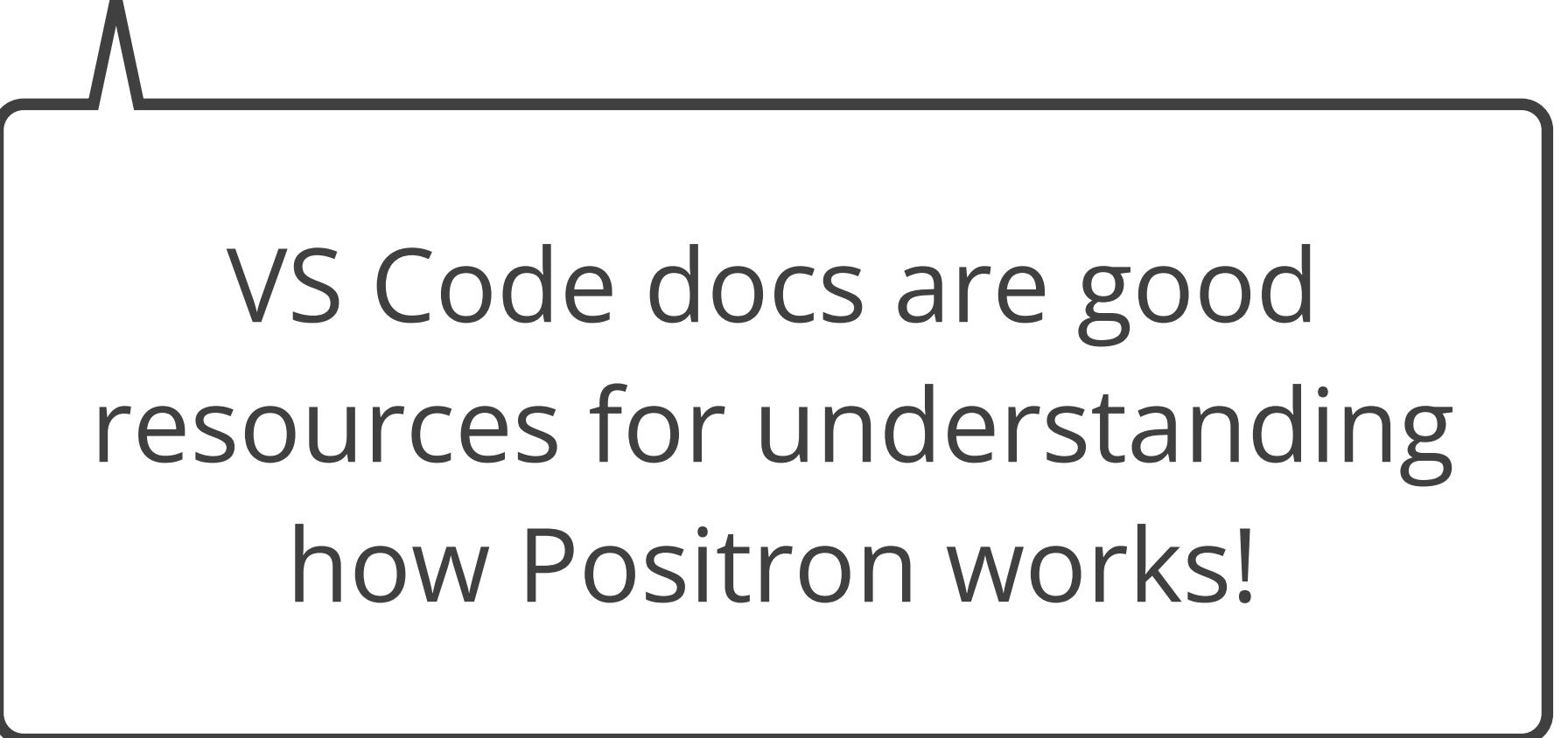
The line "editor.fontSize": 20 is highlighted with a red box.
- Bottom Status Bar:** main*, /opt/homebrew/bin/python3, 0 △ 0, Quarto: 1.8.11, Ln 2, Col 2, Spaces: 4, UTF-8, LF, JSON with Comments



Learn more

VS Code docs: User and workspace settings

code.visualstudio.com/docs/configure/settings



VS Code docs are good resources for understanding how Positron works!

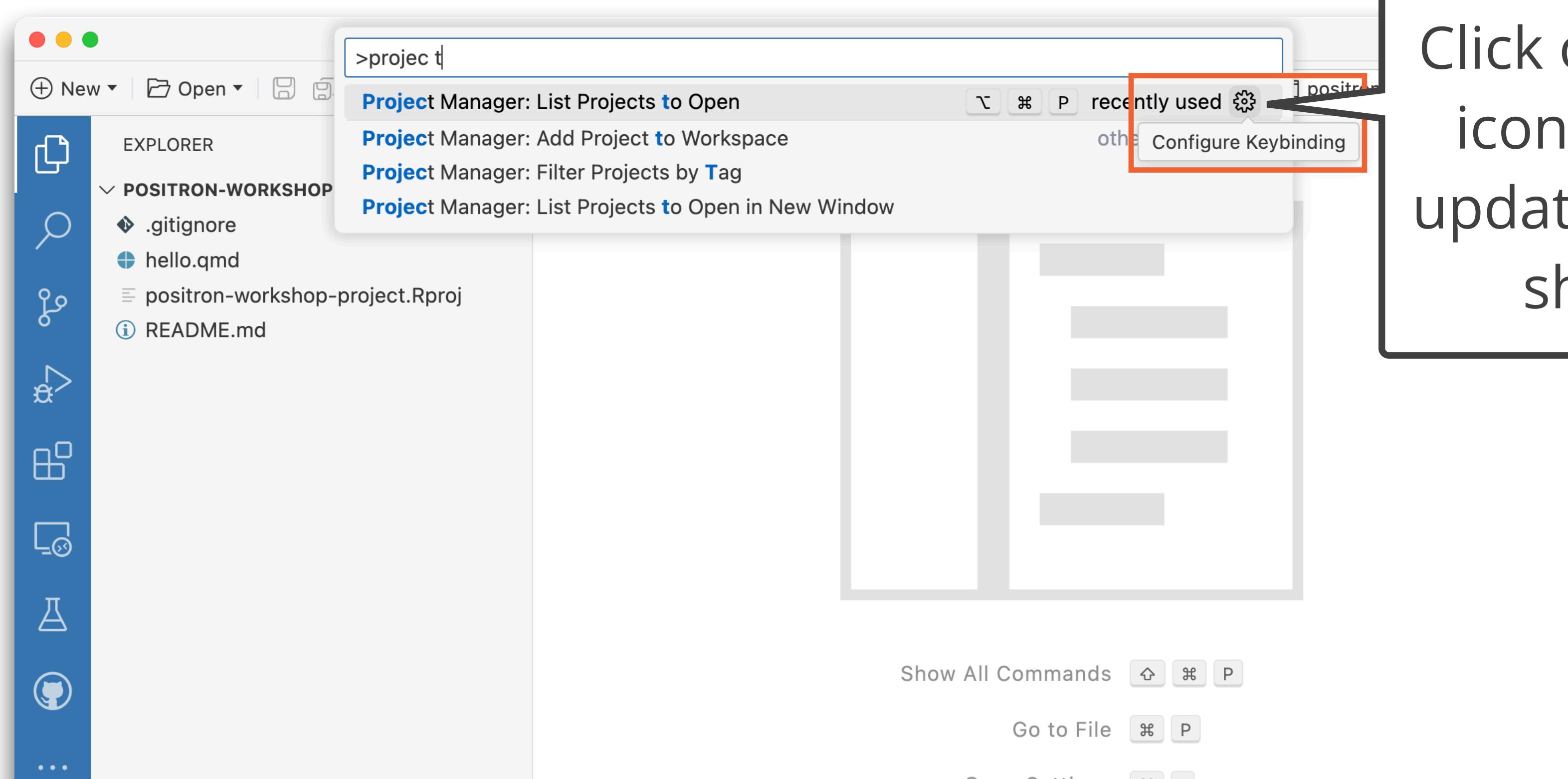
Keyboard shortcuts

Keyboard shortcuts

- Positron lets you perform most tasks directly from the keyboard.
- The Keyboard Shortcuts editor lists all available commands with and without keyboard shortcuts, and enables you to change, remove, or reset their keyboard shortcuts using the available actions.
- You can open the Keyboard Shortcuts editor:
 - From the Command Palette: *Preferences: Open Keyboard Shortcuts*
 - From the top menu: *Code > Settings > Keyboard Shortcuts*

Keyboard shortcuts for Command Palette actions

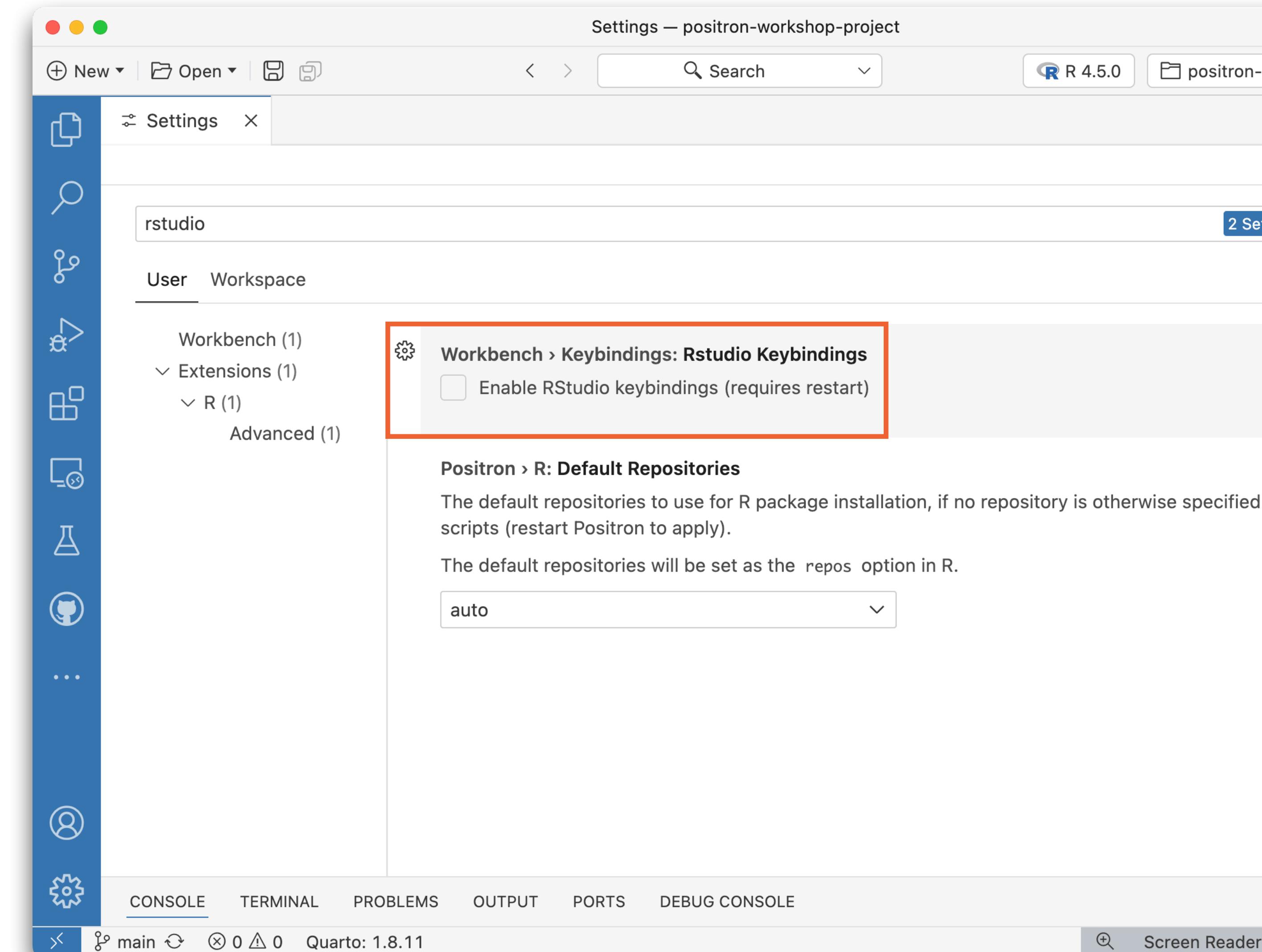
You can add a keyboard shortcut or customize the existing keyboard shortcut for any action you can take from the Command Palette.



Click on the gear icon to add or update keyboard shortcut.

Your RStudio keybindings in Positron

- You can enable RStudio keybindings in Positron with the Enable RStudio keybindings option.
- This might overwrite some of Positron's keybindings, so you might then wish to update / remove some newly enabled RStudio keybindings.



Learn more

VS Code docs: Keyboard shortcuts for Visual Studio Code

code.visualstudio.com/docs/configure/keybindings

Layout

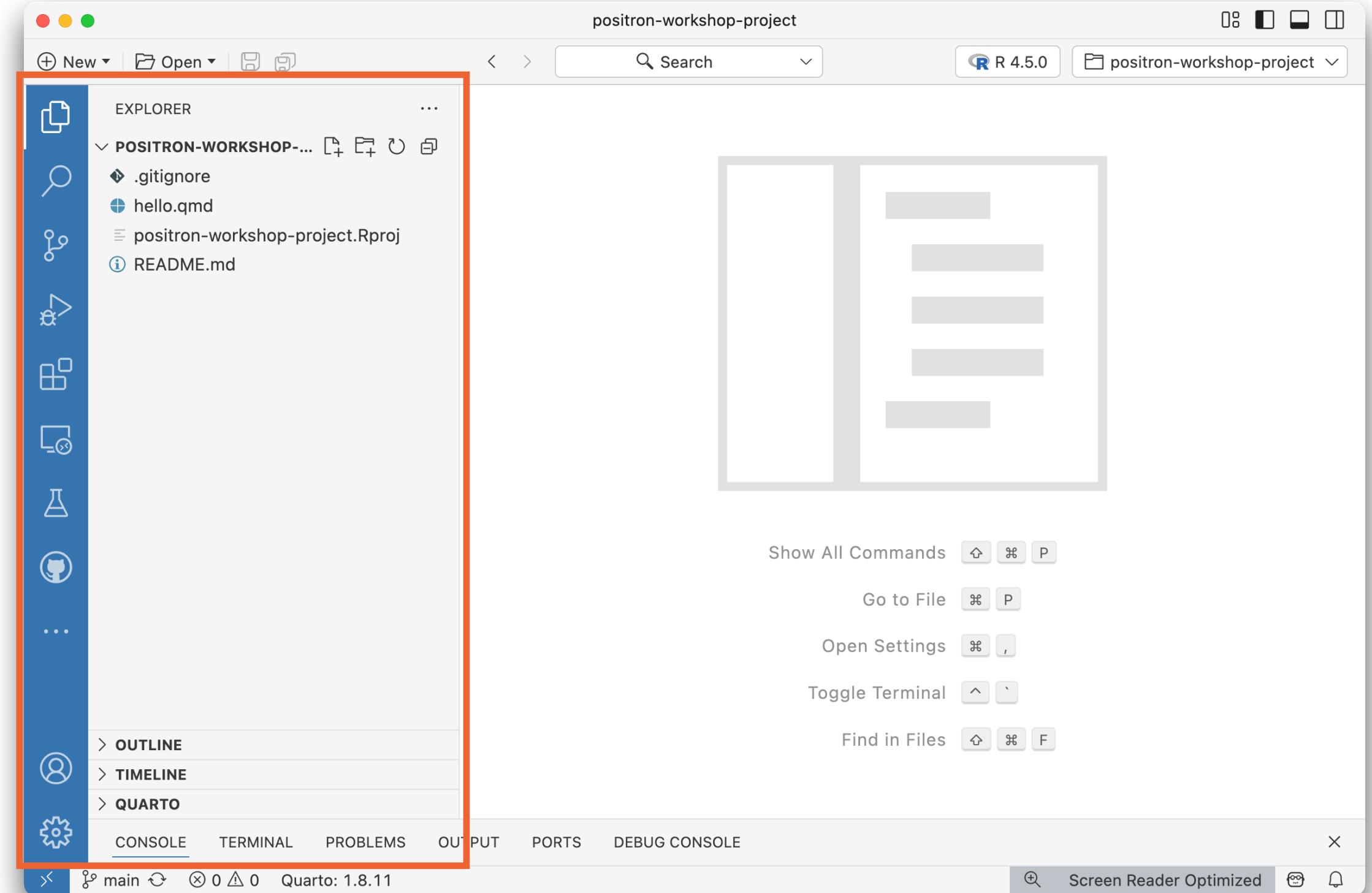
Primary side bar

- By default, the Primary Side Bar is located on the left and shows views such as the Explorer, Search, and Source Control views.
- Contents of the Primary Side Bar is controlled by what is selected in the Activity Bar.

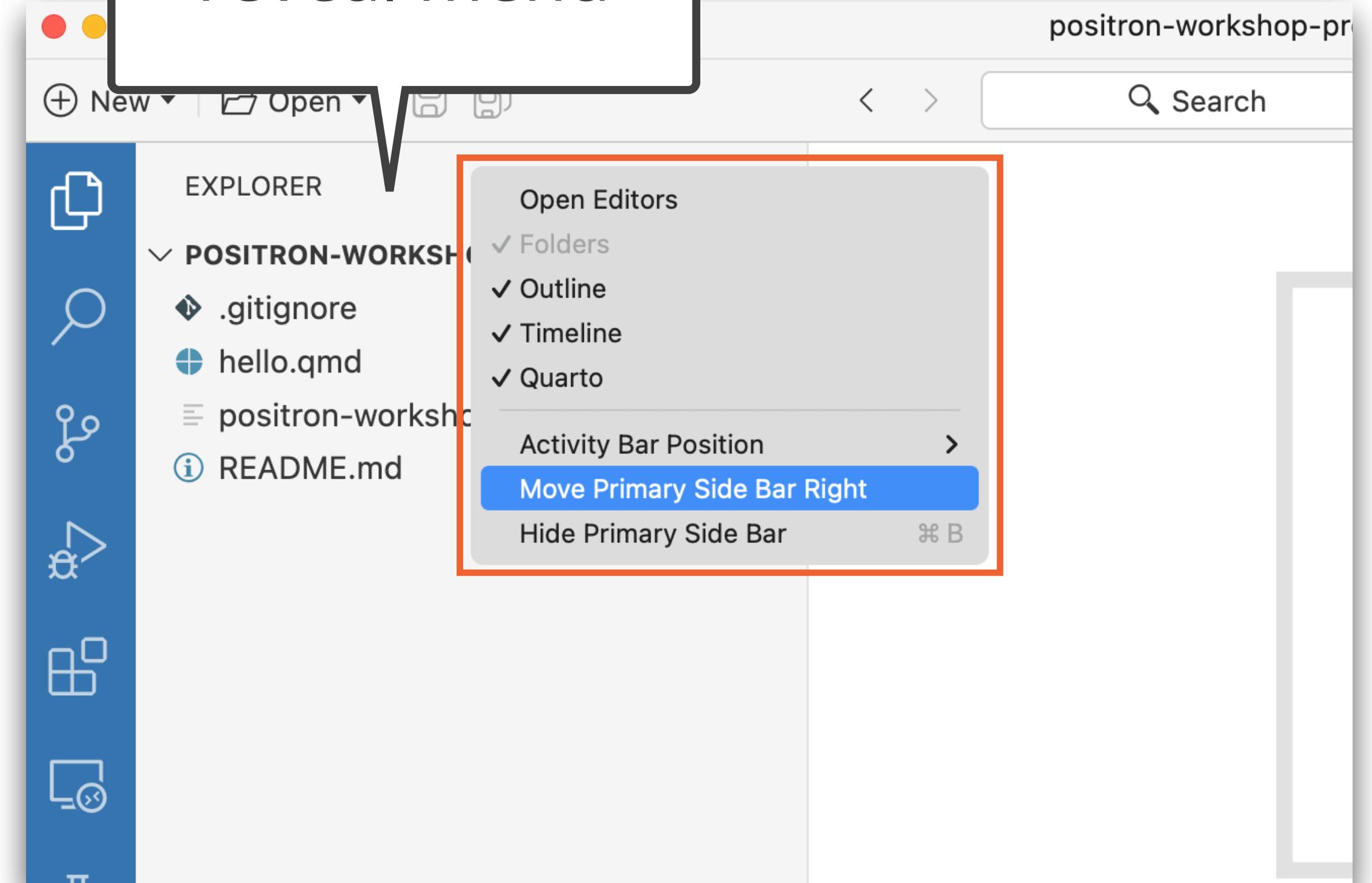
Demo

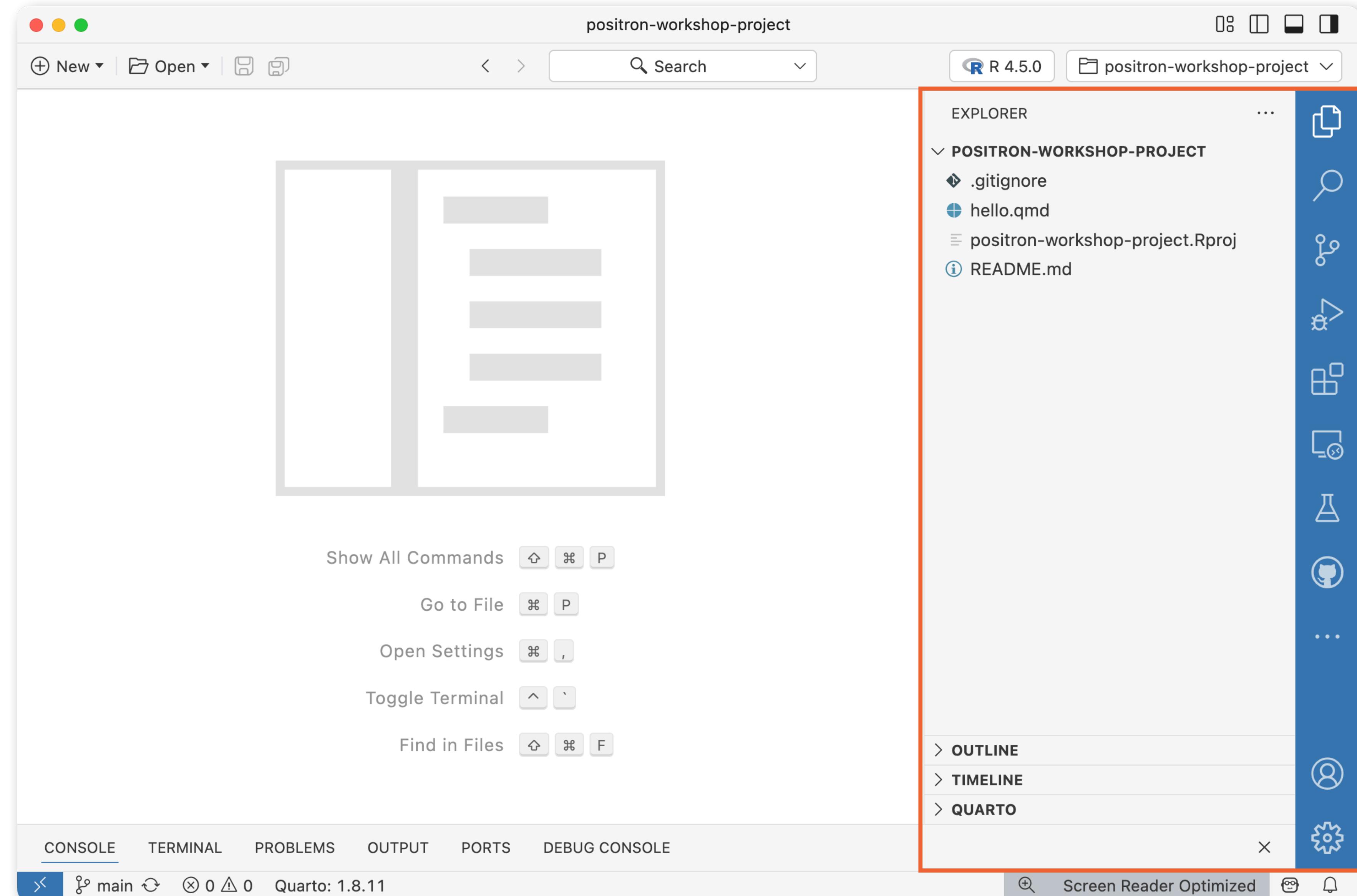
Customizing layout

- Move the primary side bar to the right.
- Move the activity bar to the top.

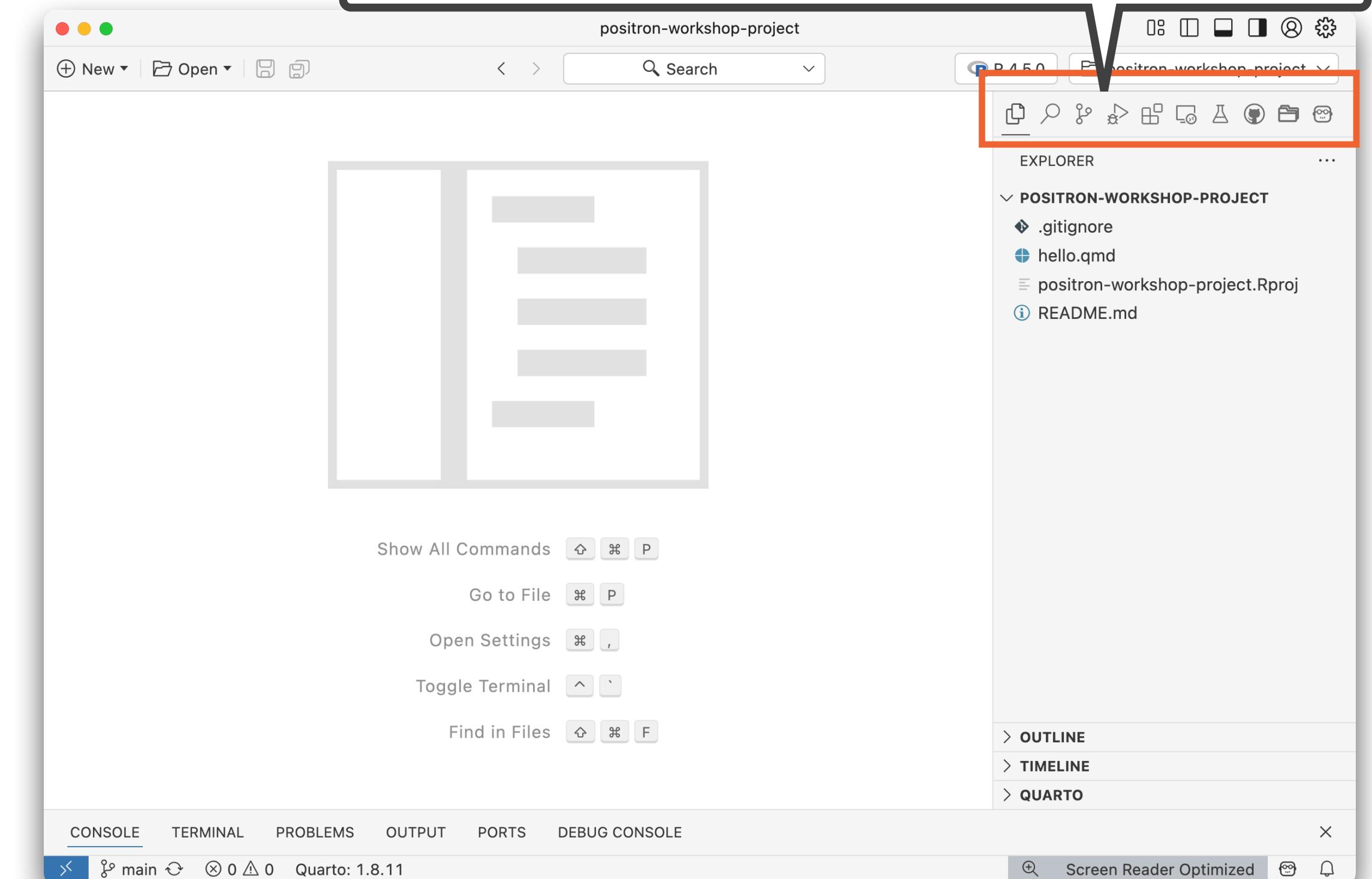
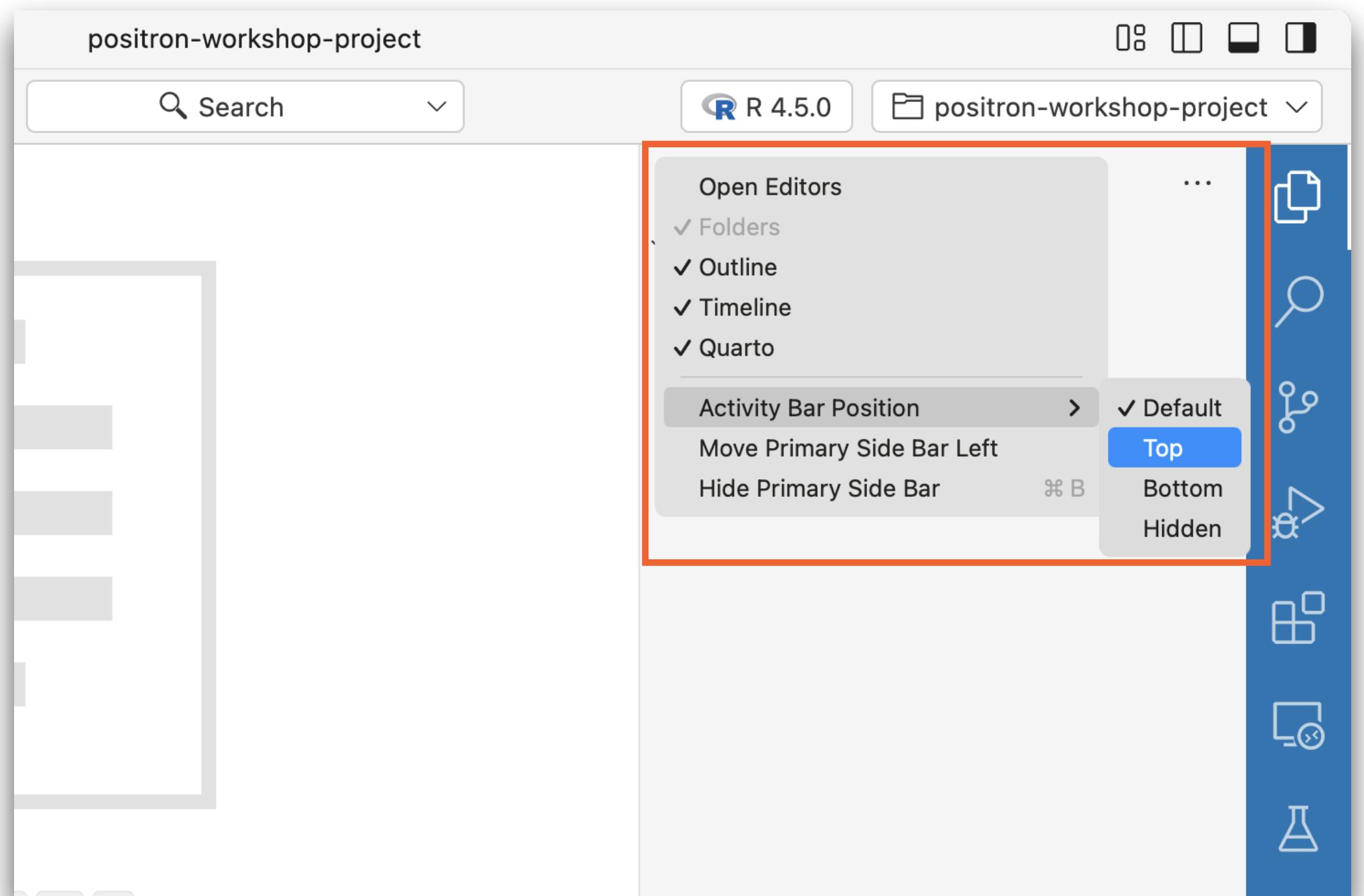


Right click to reveal menu



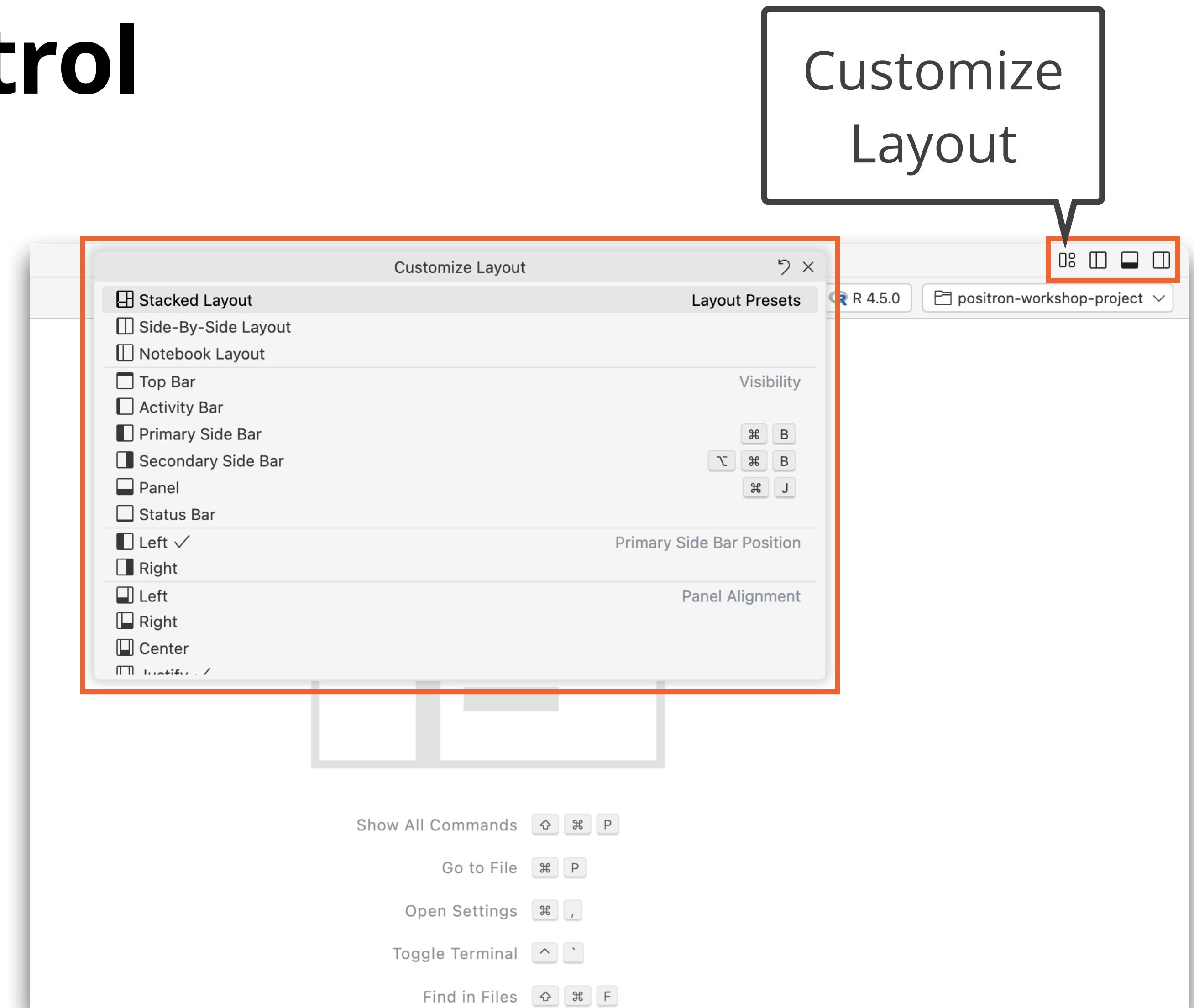


Activity Bar on top,
and not blue anymore.



Customize layout control

- The title bar has buttons to toggle the visibility of UI elements.
- The rightmost button brings up the Customize Layout dropdown that gives you finer control over the visibility and layout of various UI elements.





Teaching tip

Quarto
document /
editor —
Full height

The screenshot shows a side-by-side layout with two main panes. The left pane is the Quarto document editor, which contains:

- A code block for generating a QMD file:

```
title: "Hello, Quarto"
format: html
editor: visual
```

- A code block for loading packages:

```
1 #| label: load-packages
2 #| include: false
3 library(tidyverse)
4 library(palmerpenguins)
```

- A section titled "Meet Quarto" with a brief description and a link to <https://quarto.org>.
- A section titled "Meet the penguins" featuring an illustration of three penguins labeled CHINSTRAP!, GENTOO!, and ADÉLIE!.
- A code block for plotting penguin data:

```
1 #| label: plot-penguins
2 #| warning: false
3 #| echo: false
```

The right pane is the Positron R console, which contains:

 - A "SESSION" tab with the R version R 4.5.1.
 - A "CONSOLE" tab showing R code and its output, including a warning message about removed rows:

```
+ scale_color_manual(values = c("darkorange","purple","cyan4")) +
+   labs(
+     title = "Flipper and bill length",
+     subtitle = "Dimensions for penguins at Palmer Station LTER",
+     x = "Flipper length (mm)", y = "Bill length (mm)",
+     color = "Penguin species", shape = "Penguin species"
+   ) +
+   theme_minimal()
Warning message:
Removed 2 rows containing missing values or values outside the scale range
(`geom_point()`).
```

 - A "VARIABLES" tab showing the "penguins" dataset.
 - A "DATA" tab showing the "penguins" dataset with columns: species, island, bill length, flipper length, etc.
 - A "PLOTS" tab showing a scatter plot titled "Flipper and bill length" with "Dimensions for penguins at Palmer Station LTER". The x-axis is "Flipper length (mm)" and the y-axis is "Bill length (mm)". The plot uses color and shape to represent penguin species: Adelie (orange dots), Chinstrap (purple triangles), and Gentoo (teal squares).

Side-by-side
layout

Console —
Prompt visible
from the back of
the classroom

Variables and
data

Plots — that can
be popped out
into the editor /
new window for
inspection

Learn more

VS Code docs: Custom Layout

code.visualstudio.com/docs/configure/custom-layout

Extensions

Extensions

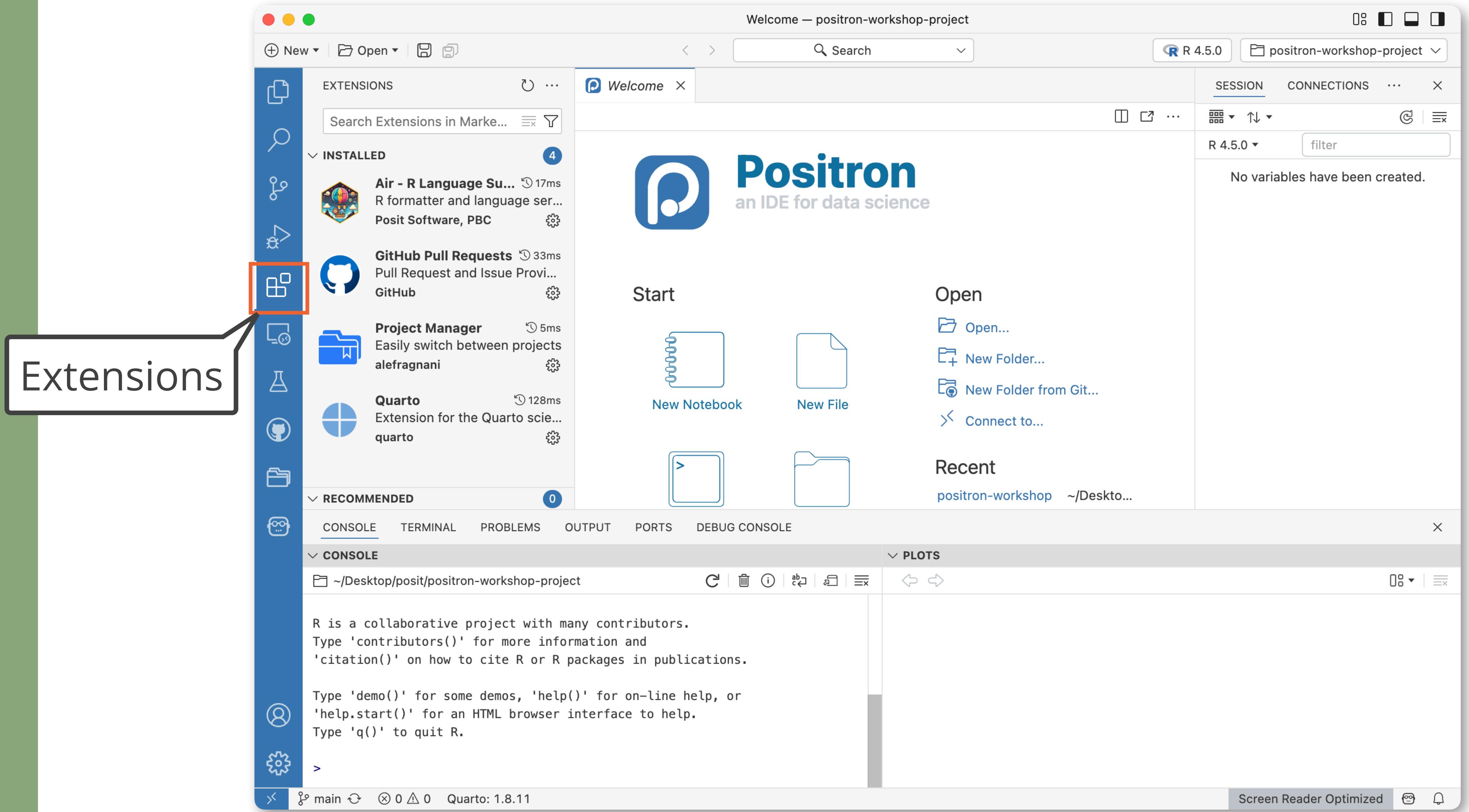
- Extensions let you add languages, debuggers, tools, themes, etc. to Positron to support your development workflow (just like in VS Code).
- Your Positron and VS Code extensions are separate and won't interfere with each other.
- Unlike in VS Code, you install Positron extensions from [Open VSX](#).

3m 00s

Your turn

Install extension

- Click on the Extensions icon from the Activity Bar in Positron.
- If you haven't yet done so, search for and install the Shiny, or Project Manager extension.
- If you have already installed it, or if you would like to install one more extension, find and install the Solarized & Selenized extension.
 - Use the Command Palette to set the theme of your Positron window to a Solarized or Selenized theme of your choice. **Hint:** Start typing *theme* in the Command Palete.



Welcome — positron-workshop-project

EXTENSIONS: MARKETPLACE

Solarized

Solarized & Selenized... ⚡ 1K ★ 5
A Solarized and Selenized co...
santoso-wijaya [Install](#)

Solarized Sunset Th... ⚡ 1K
A flat theme based on Solar...
jansim [Install](#)

Better Solarized ⚡ 12K
A Better Solarized theme for ...
Ed Heltzel [Install](#)

Solarized Sharp ⚡ 3K
muhammad-sammy [Install](#)

Solarized High Contr... ⚡ 96
Sepia-toned, high-contrast li...
tiny [Install](#)

SESSION CONNECTIONS ... X

R 4.5.0 positron-workshop-project

No variables have been created.

Start

New Notebook New File

Open

Open... New Folder... New Folder from Git... Connect to...

Recent

positron-workshop ~/Desktop...

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE

CONSOLE

~ /Desktop/posit/positron-workshop-project

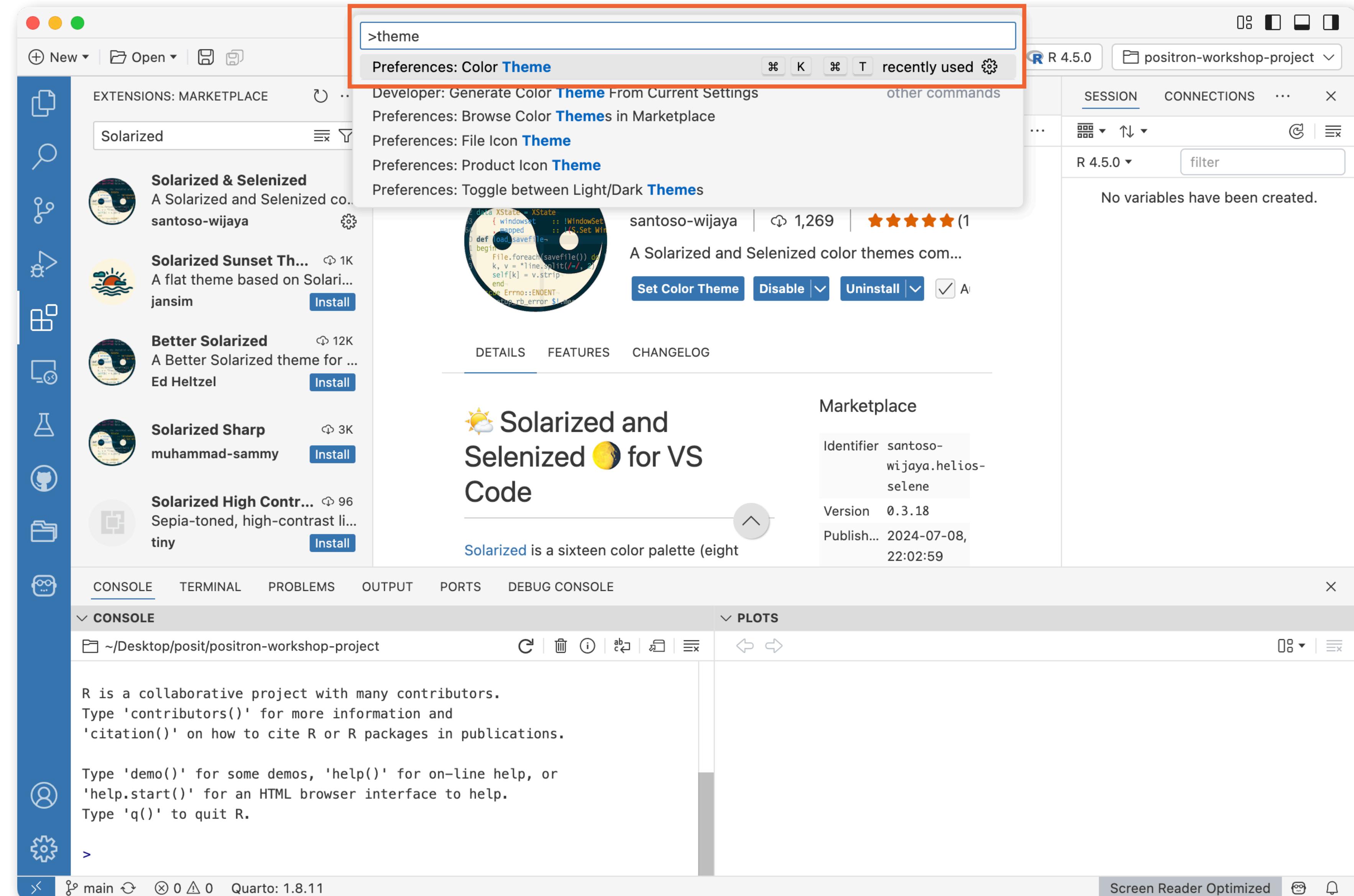
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

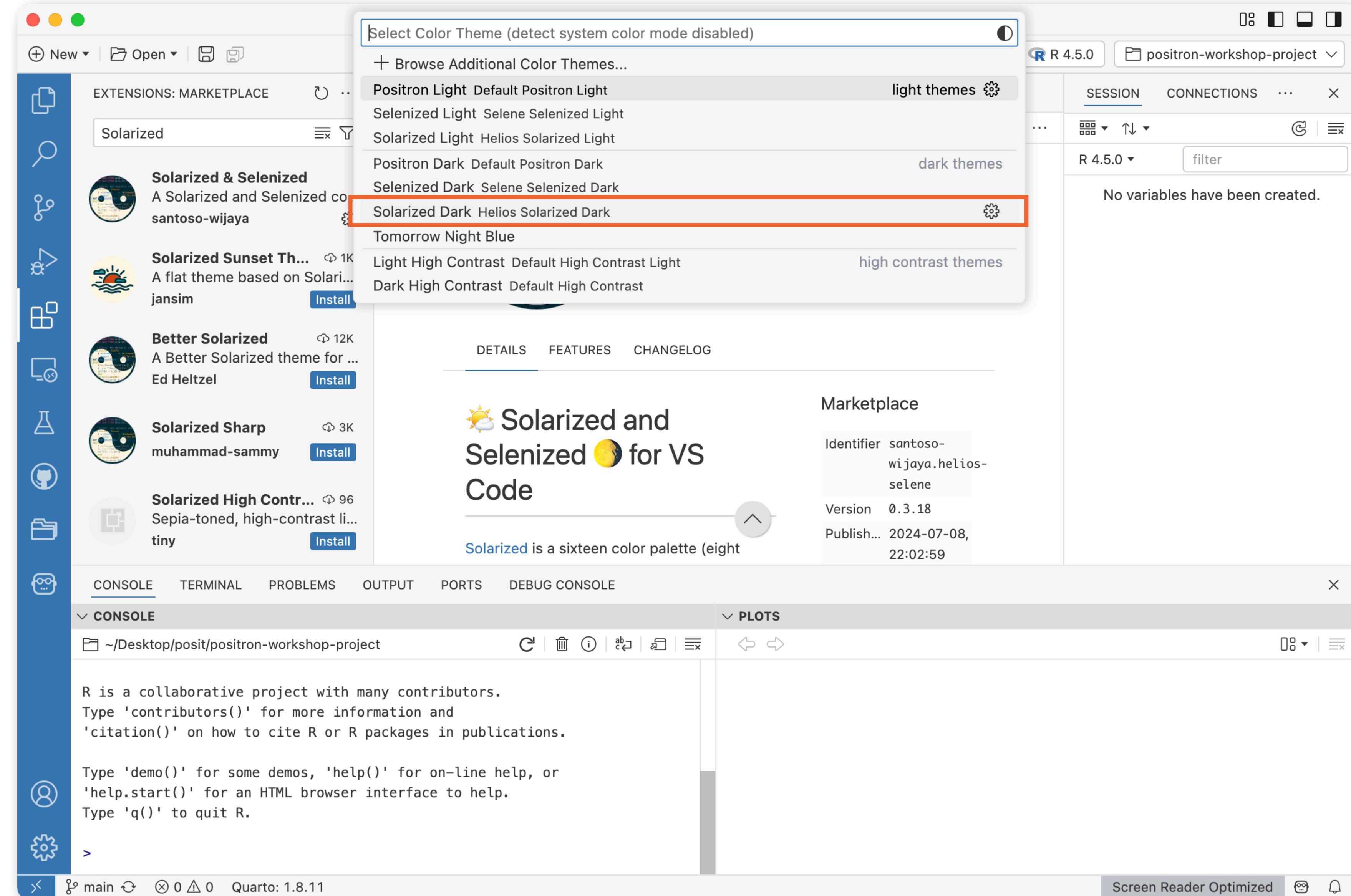
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

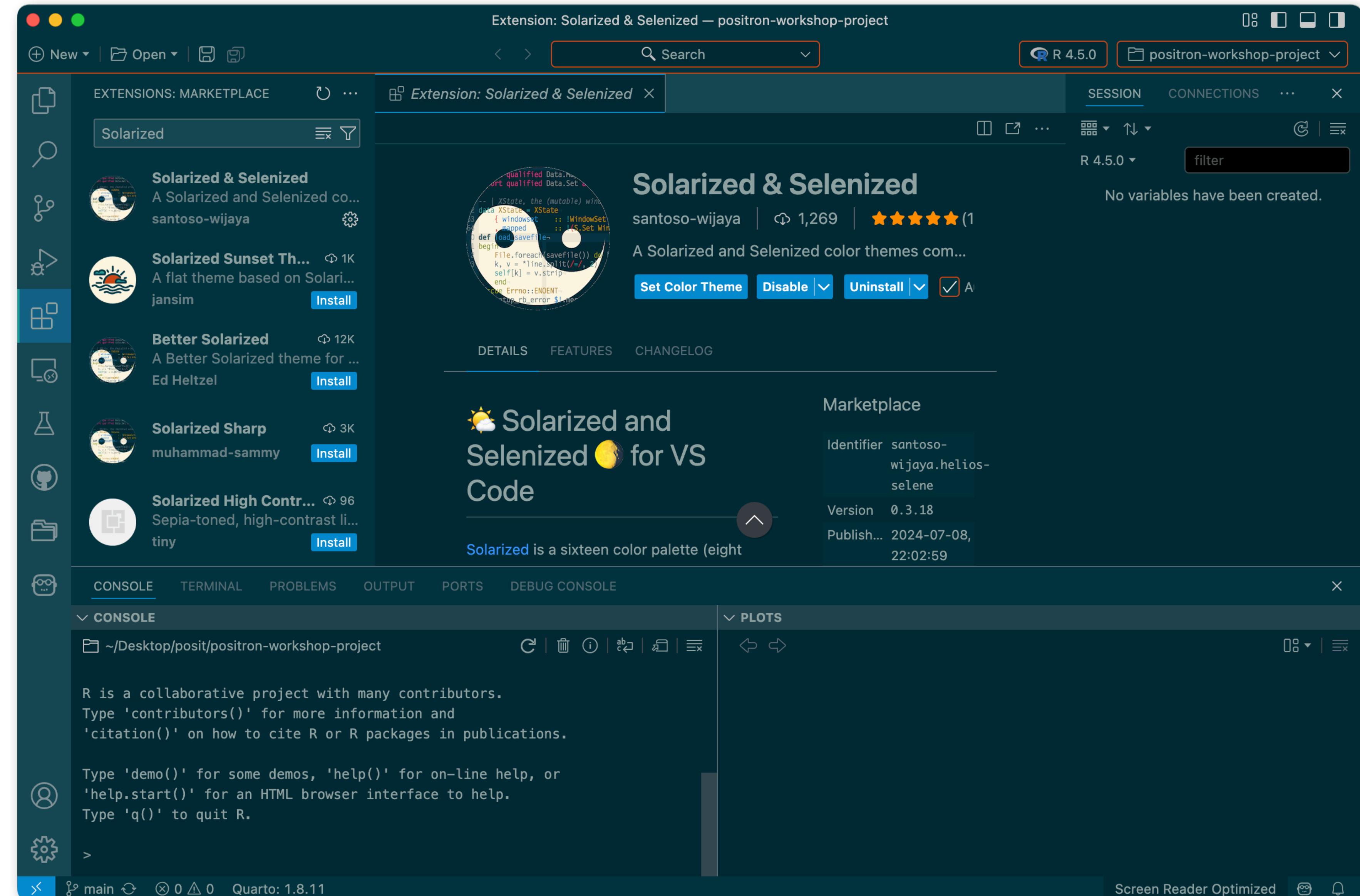
>

main ↻ ⊗ 0 △ 0 Quarto: 1.8.11 Screen Reader Optimized

The screenshot shows the Positron IDE interface. The top navigation bar displays the title "Welcome — positron-workshop-project". Below the title, there are tabs for "SESSION" (R 4.5.0), "CONNECTIONS", and a dropdown menu for the project "positron-workshop-project". A search bar is located at the top right. On the left side, a vertical toolbar contains icons for file operations, search, connections, and other tools. The main workspace features a central "Welcome" panel with the Positron logo and the text "an IDE for data science". To the left of the welcome panel is a list of extensions from the Marketplace, with the "Solarized" extension highlighted by a red box. The "Start" section provides options to "New Notebook" or "New File". On the right, there are sections for "Open" (with links to "Open...", "New Folder...", "New Folder from Git...", and "Connect to...") and "Recent" (listing the "positron-workshop" project). At the bottom, there are tabs for "CONSOLE", "TERMINAL", "PROBLEMS", "OUTPUT", "PORTS", and "DEBUG CONSOLE". The "CONSOLE" tab is active, showing the R environment setup. The "PLOTS" tab is also visible. The bottom status bar includes a "Screen Reader Optimized" button and system status indicators like battery level and signal strength.







Solarized
Dark
theme

Built-in extensions

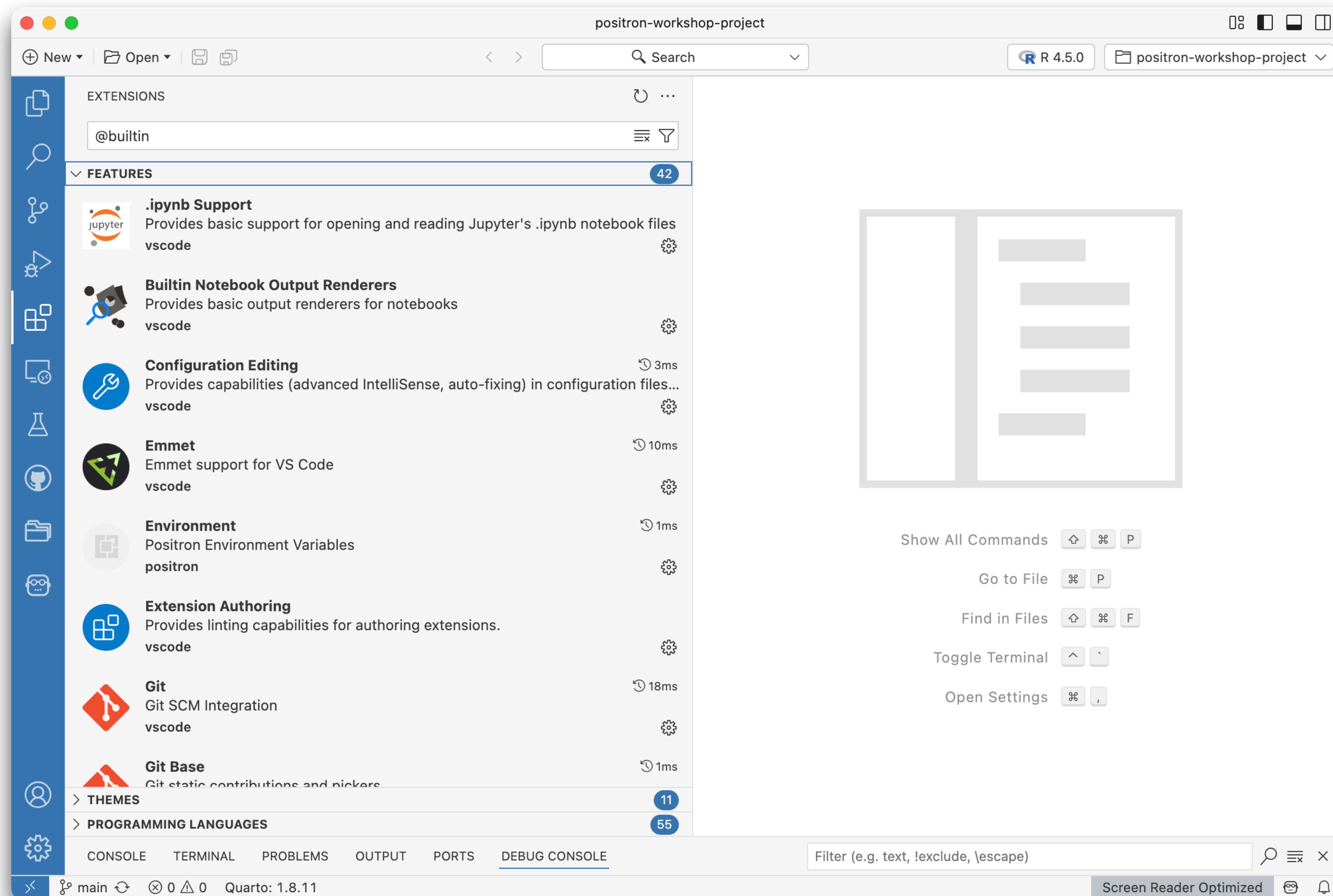
- Built-in extensions (e.g., R, Positron Python, and Markdown Language Basics) are installed with Positron.
- They can only be updated by updating Positron.
- They can be seen in the extensions view with `@builtin`.

Demo

Built-in extensions

- Click on the Extensions icon in the Activity Bar.
- Type `@builtin` in the Extension search bar.
- Review the built-in extensions.

Built-in extensions



Bundled extensions

- Bundled extensions (e.g., Quarto, Air, GitHub Pull Requests) are installed with Positron.
- They can be updated independently of updating Positron.

Other extensions

- Any other extensions need to be installed from Open VSX.
- Popular extensions you might consider installing are Project Manager, Shiny, Rainbow CSV, indent-rainbow, etc.
- You can also install Extension Packs; a collection of packs curated by users for specific tastes, tasks, or languages.

5m 00s

Your turn

Install the ugliest theme extension

- Find the ugliest theme extension you can.
- Install the extension and apply the theme.
- **Note:** You might get a warning about whether you trust the publisher of a theme you choose. Read the warning carefully. The decision is yours!

Extension: Dracula Theme Official — positron-workshop-project

EXTENSIONS: MARKETPLACE

theme

Solarized & Selenized
A Solarized and Selenized colo...
santoso-wijaya

Dracula Theme Official
The official Dracula Theme: a ...
dracula-theme Installing

base9-vscode
base9-theme Install

Wildberries
A dark theme for purple lovers
wildberries-theme Install

Material Theme (But I ...
A Apache-2 licensed fork of M...
t3dotgg Install

Cobalt2 Theme O...
Official theme by Wes Bos.
wesbos Install

XD Theme
A theme because my settings ...
jeff-hykin Install

OpenSumi Default Th...
The default OpenSumi themes
opensumi Install

MSDevStudio Default...
745

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE

Search

R 4.5.0 positron-workshop-project

Extension: Dracula Theme Official

Dracula Theme Official

dracula-theme | 158,746 | ★★★★★ (2)

The official Dracula Theme: a dark theme for many editors, shells, and more. ...

! Do you trust the publisher "dracula-theme"?

The extension **Dracula Theme Official** is published by dracula-theme. This is the first extension you're installing from this publisher.

dracula-theme is **not verified**.

Positron has no control over the behavior of third-party extensions, including how they manage your personal data. Proceed only if you trust the publisher.

Learn More Cancel Trust Publisher & Install

Marketplace

Identifier dracula-theme.theme-dracula

Version 2.25.1

Publish... 2020-04-06, 23:07:47

Last 2024-09-05, Released 23:29:53

Categories Themes

Resources Marketplace Repository

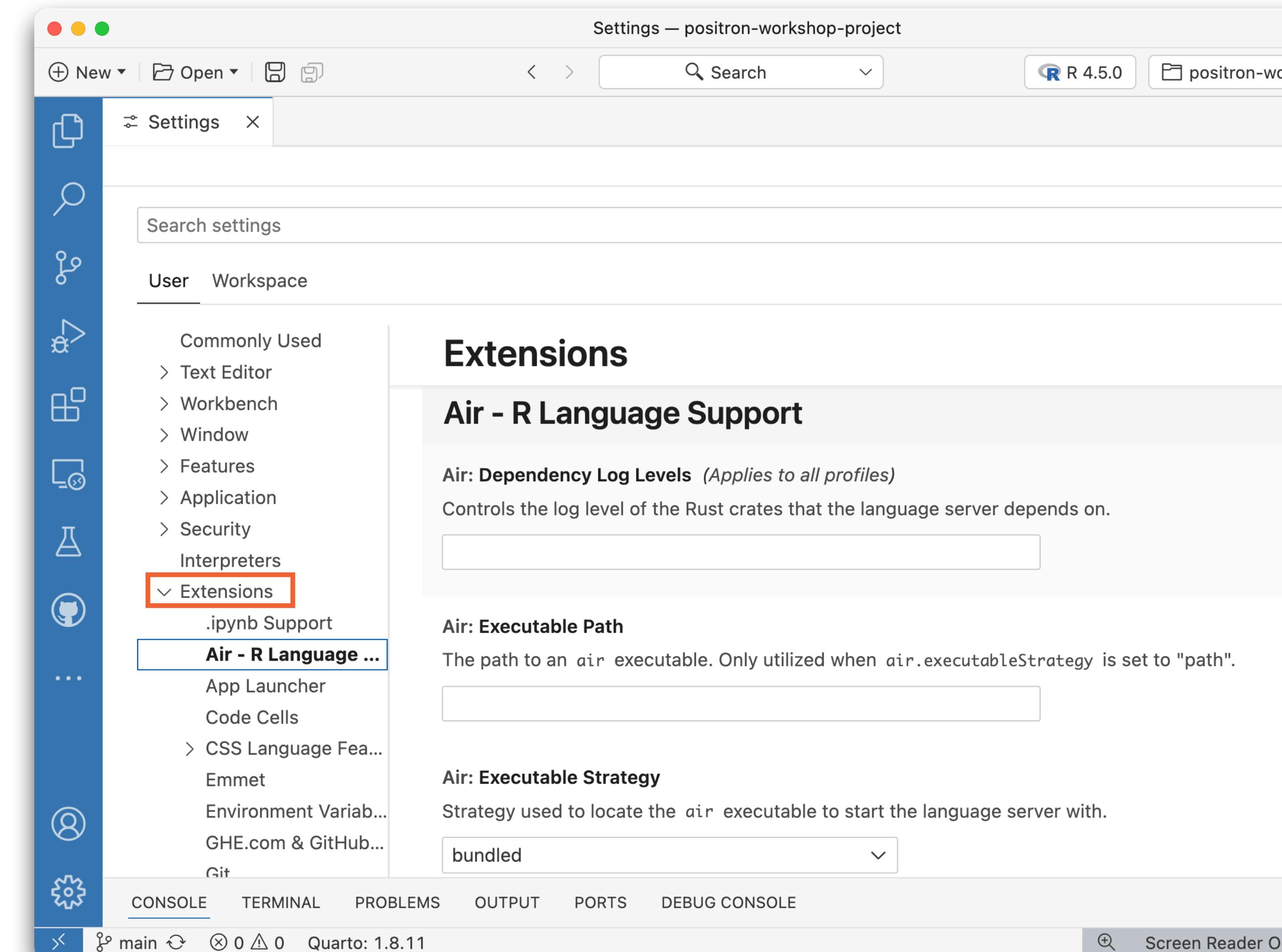
Filter (e.g. text, !exclude, \escape)

Screen Reader Optimized

main 0 △ 0 Quarto: 1.8.11

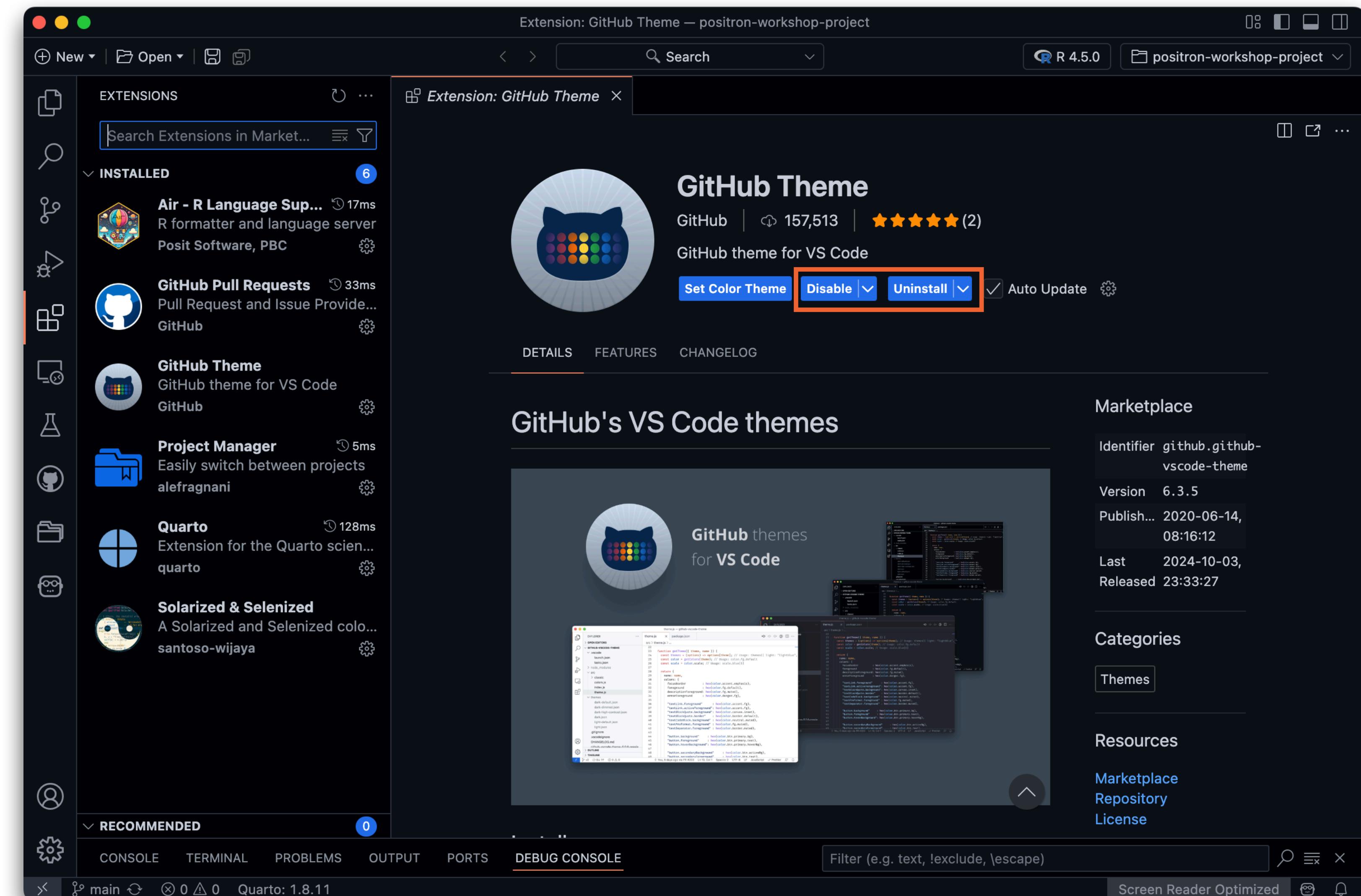
Extension settings

- Installed Positron extensions can also contribute their own settings, which you can review under the Extensions section of the Settings editor.
- These can be set at the user or workspace level as well.



A word of caution

Extensions alter the look and functionality of Positron and can conflict with each other. We recommend being selective in extensions you have enabled at any given point, and disabling or uninstalling the ones you no longer use.



Learn more

Positron docs: Extensions

positron.posit.co/Extensions.html