

Customization + extensibility

Part of "Doing data science in Positron" workshop



Settings

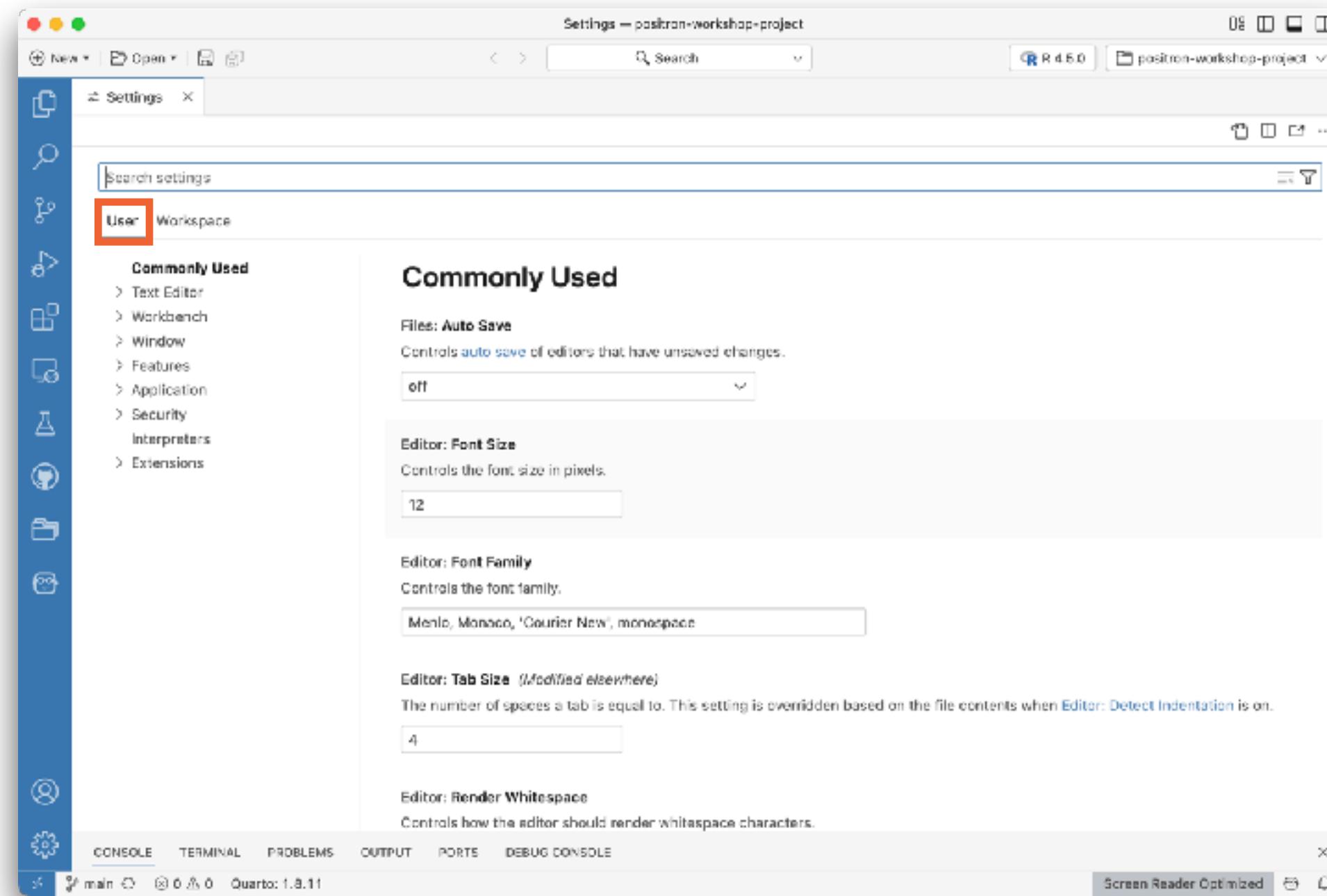
Settings

- 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 displaying the file "settings.json". The title bar says "settings.json - positron-workshop-project". The code editor shows the following JSON content:

```
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 }
```

A red box highlights the entire JSON code. A callout bubble with a black border and a blue arrow points from the text "Path to settings.json" to the red box. The status bar at the bottom of the code editor shows "main" and "/opt/homebrew/bin/python3".

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 | R 4.5.0 | positron-workshop-project

Search 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`.

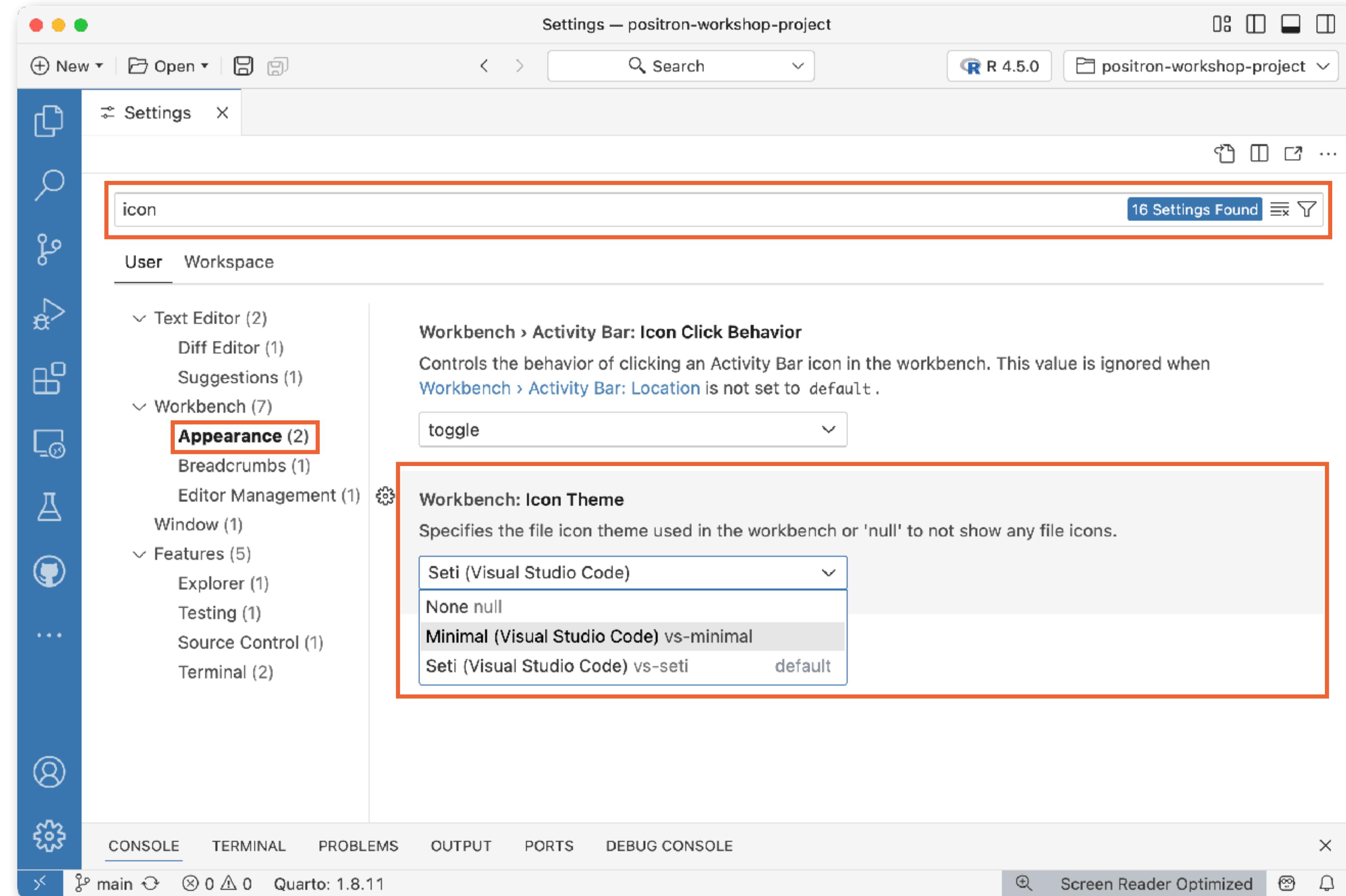
toggle

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”**.

Change to
“vs-seti”

The screenshot shows the RStudio interface with the 'hello.qmd' file open in the Quarto editor. The left sidebar displays the project structure under 'OPEN EDITORS'. The main pane shows the Quarto code:

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

The 'CONSOLE' tab at the bottom shows R startup messages.

This screenshot is similar to the first one, but a tooltip has appeared over the 'format:' line in the Quarto code. The tooltip content is:

```
~/Documents/positron/positron-workshop-project/fruit_avg_browser.R
```

The rest of the interface and code are identical to the first screenshot.

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

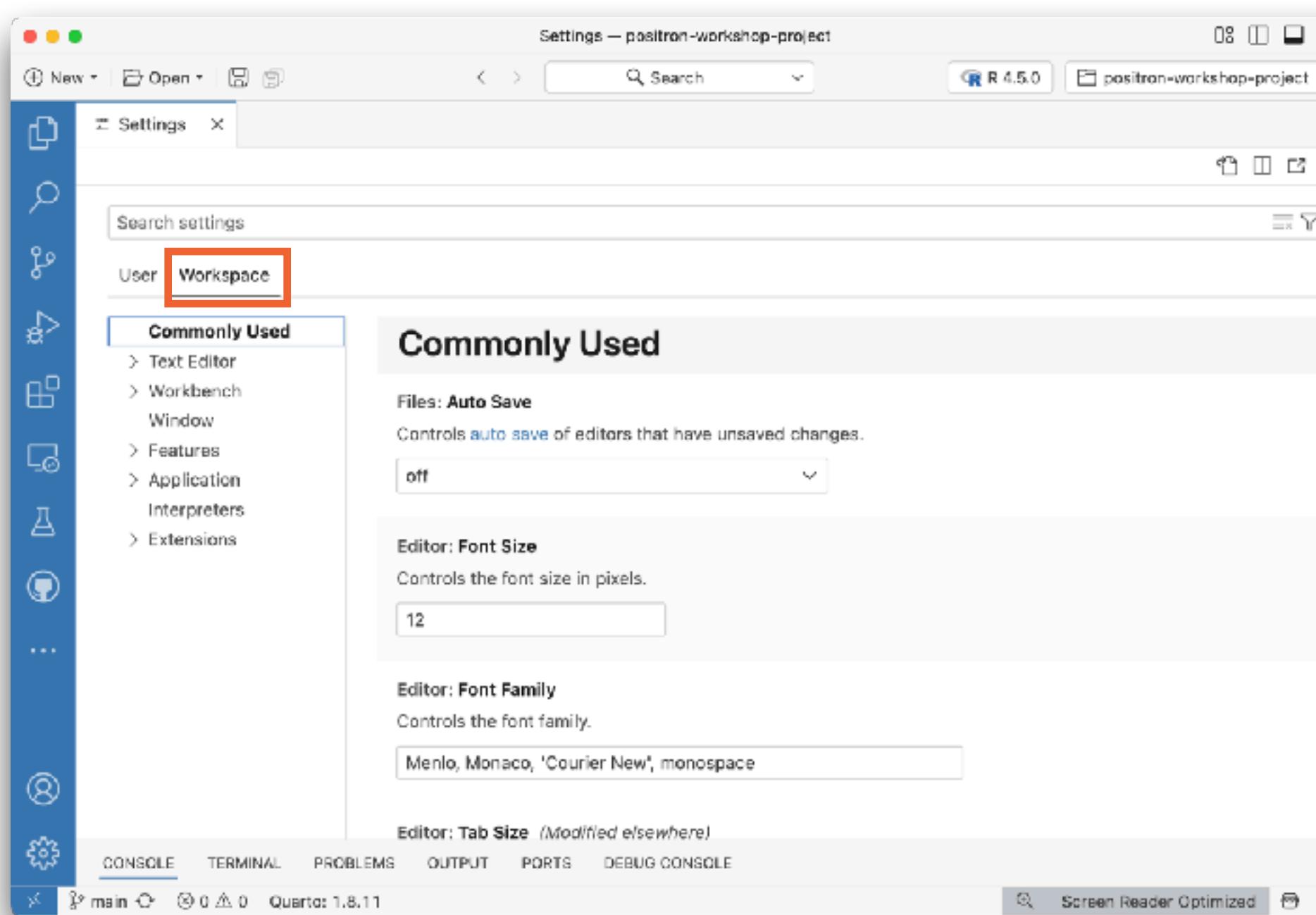
To update using the UI:

- In the Command Palette:

Preferences: Open Workspace Settings

- In the top menu:

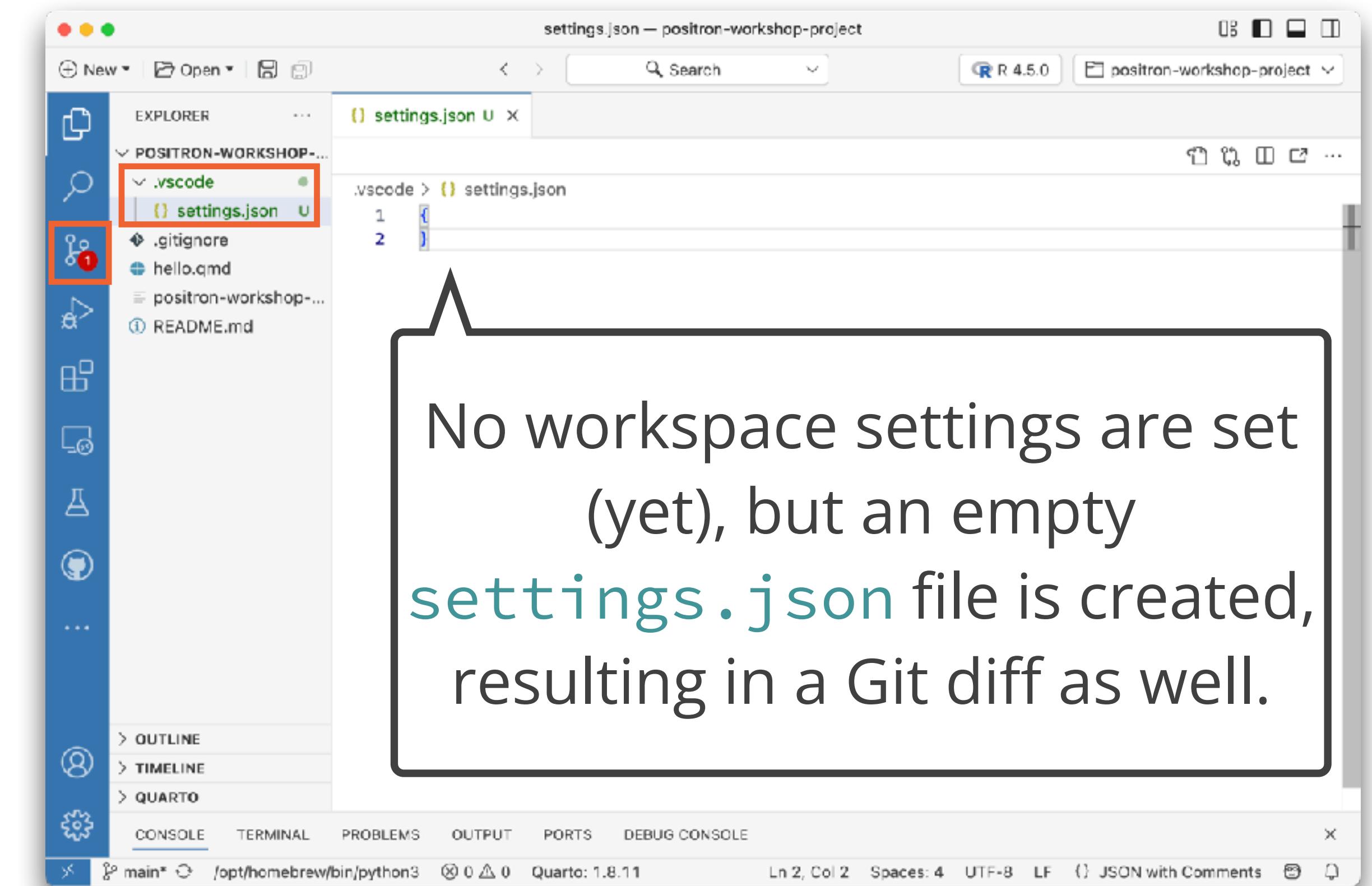
Positron > Settings... > Settings



To edit the JSON file:

In the Command Palette:

Preferences: Open Workspace Settings (JSON)

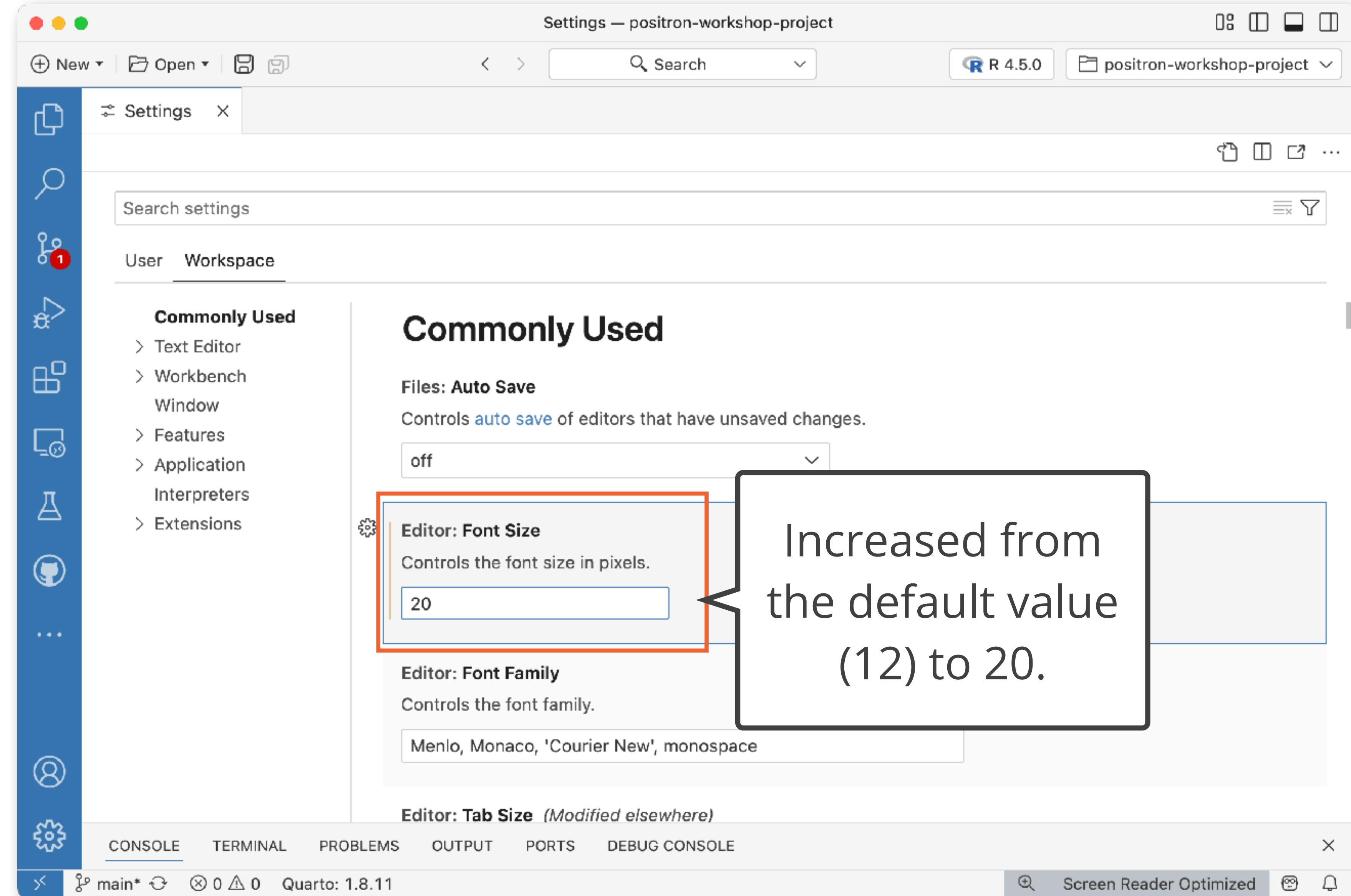


3m 00s

Your turn

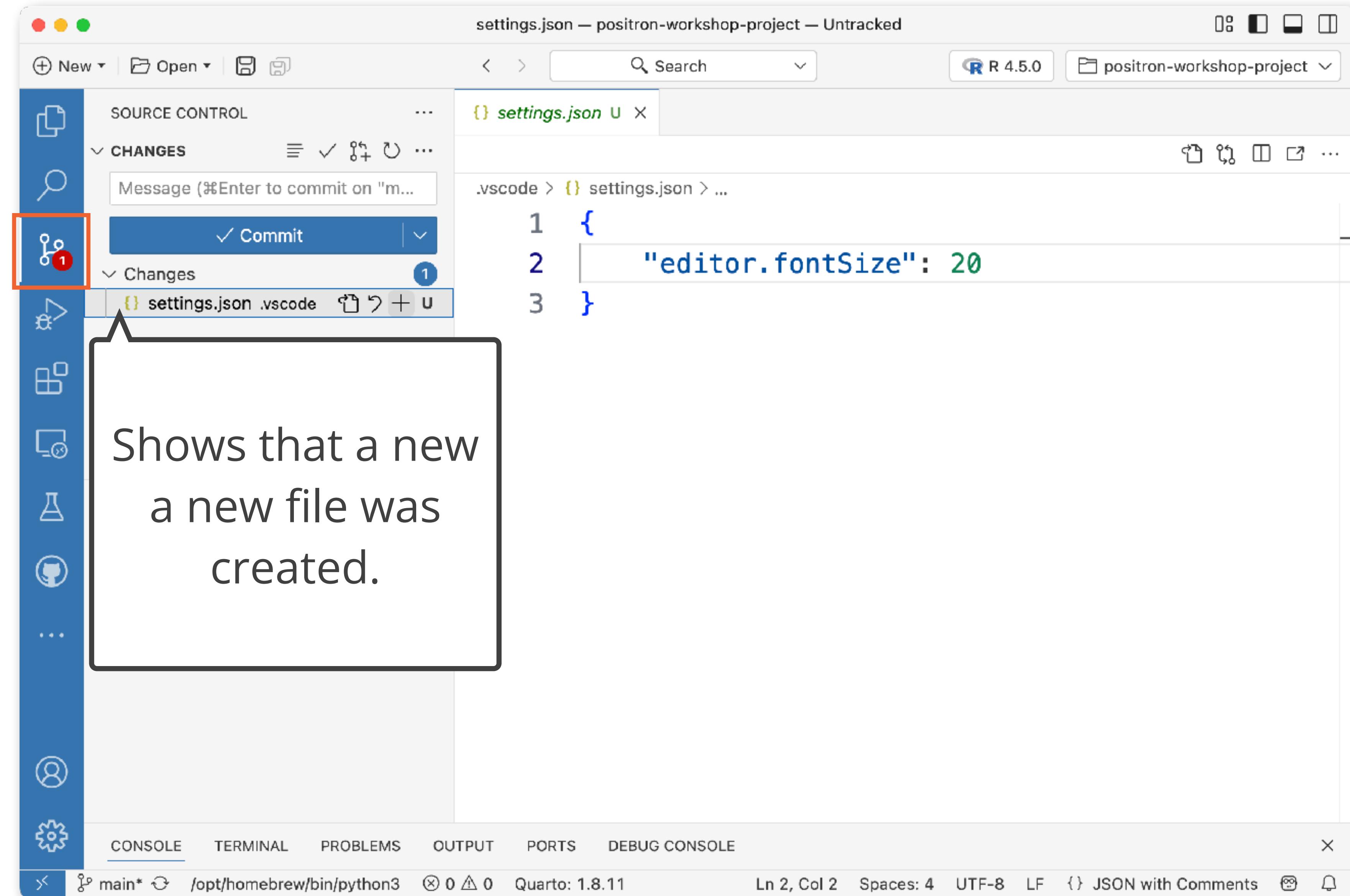
Customize workspace settings

- 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.



A screenshot of the Visual Studio Code (VS Code) interface. The title bar shows "settings.json — positron-workshop-project — Untracked". The left sidebar has a red box around the "EXPLORER" icon. The Explorer view shows a project structure with a .vscode folder containing a settings.json file, along with .gitignore, hello.qmd, and README.md files. A red box highlights the "editor.fontSize": 20 line in the settings.json file. The bottom status bar shows "main* /opt/homebrew/bin/python3" and "Quarto: 1.8.11".

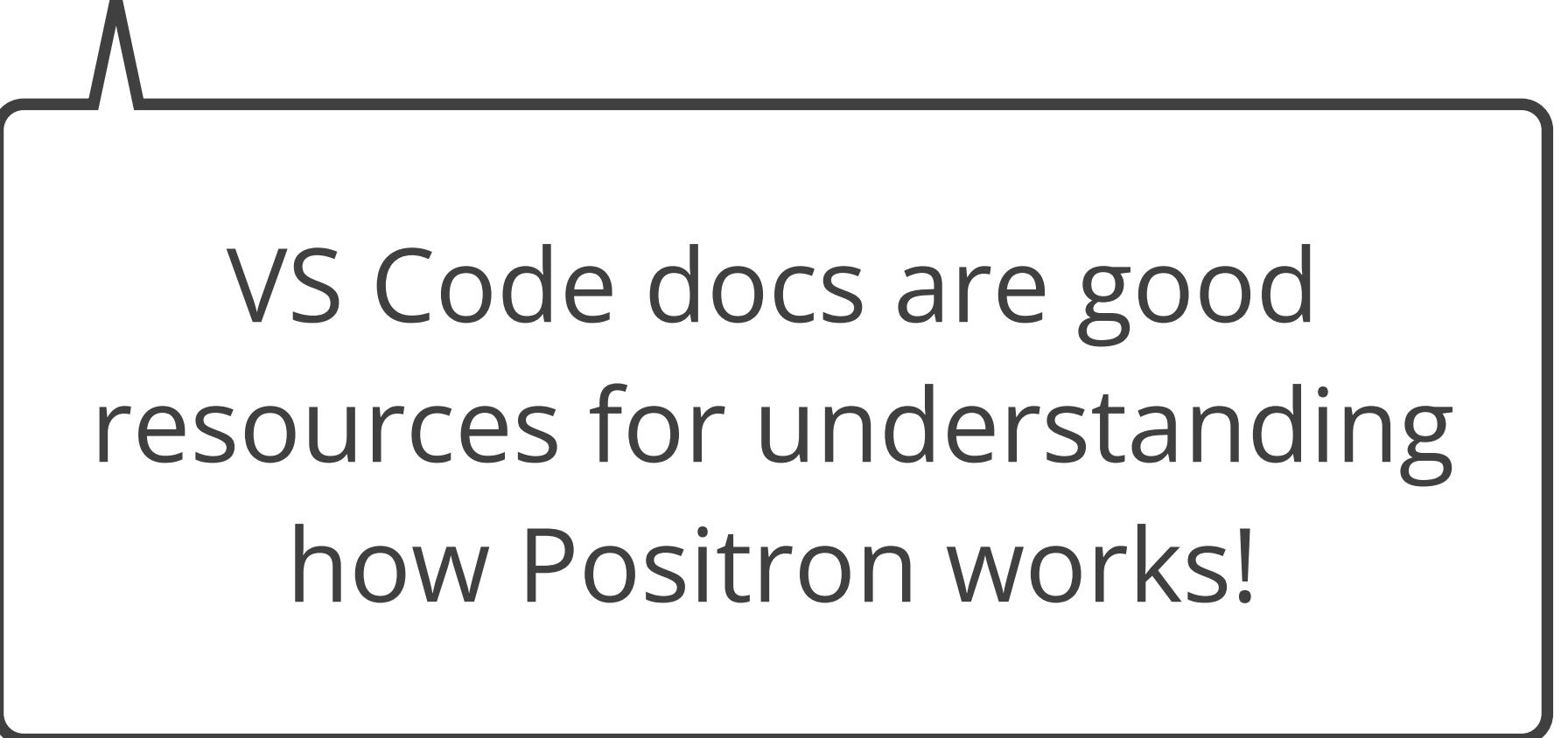
```
1 {  
2     "editor.fontSize": 20  
3 }
```



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!

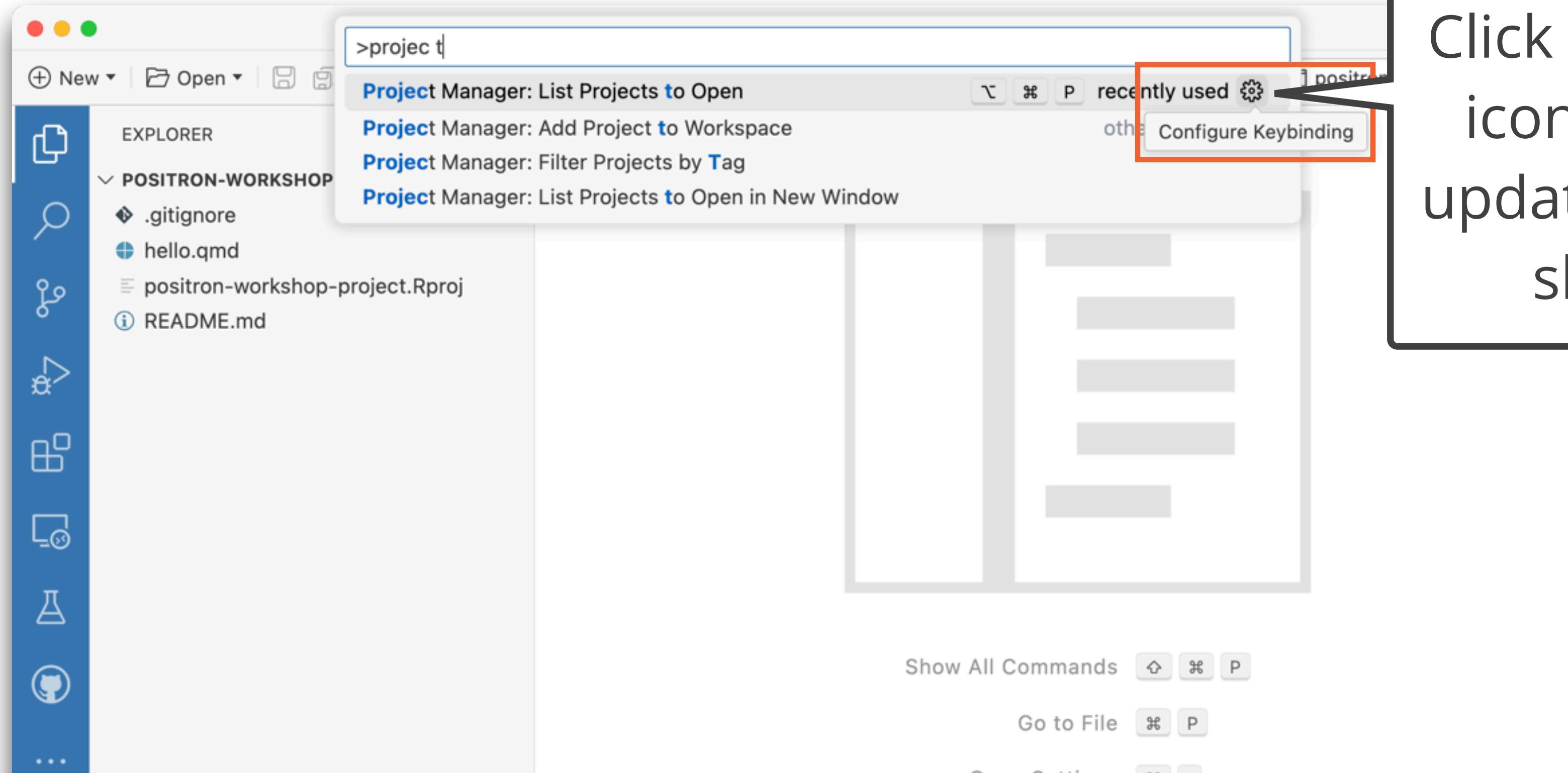
Keyboard shortcuts

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

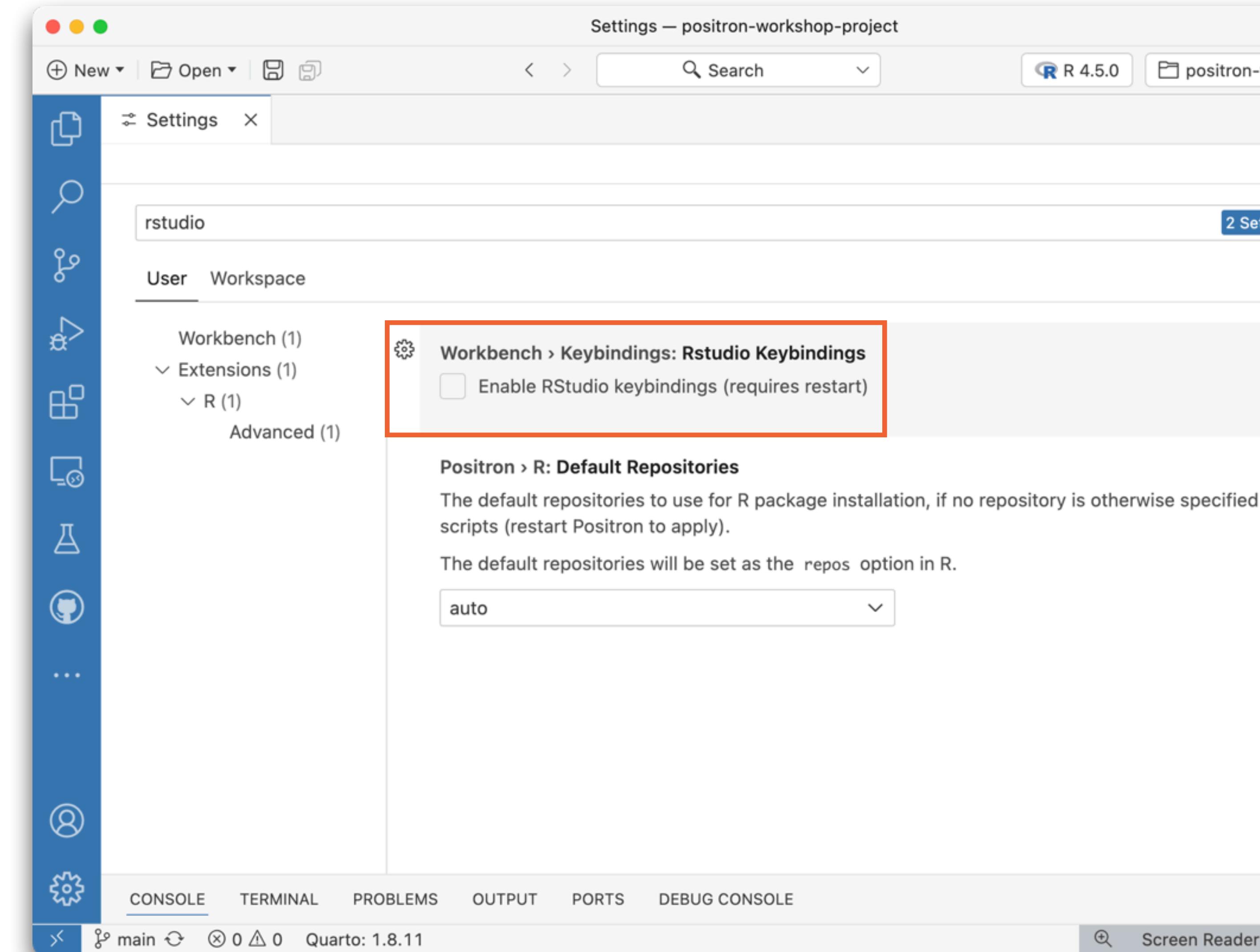
You can add a keyboard shortcut or customize the existing keyboard shortcut for any action you can take from the Command Palette.



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

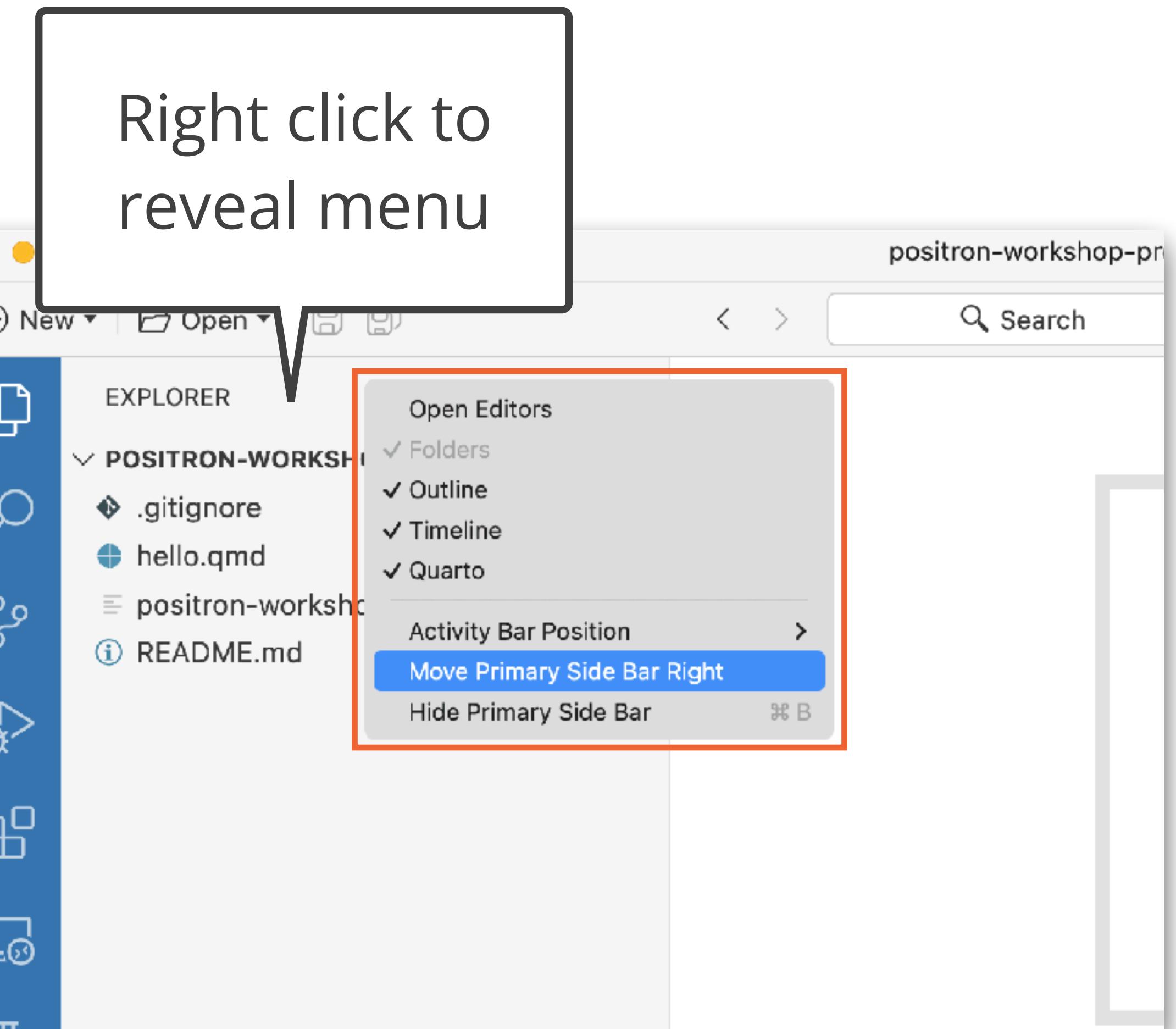
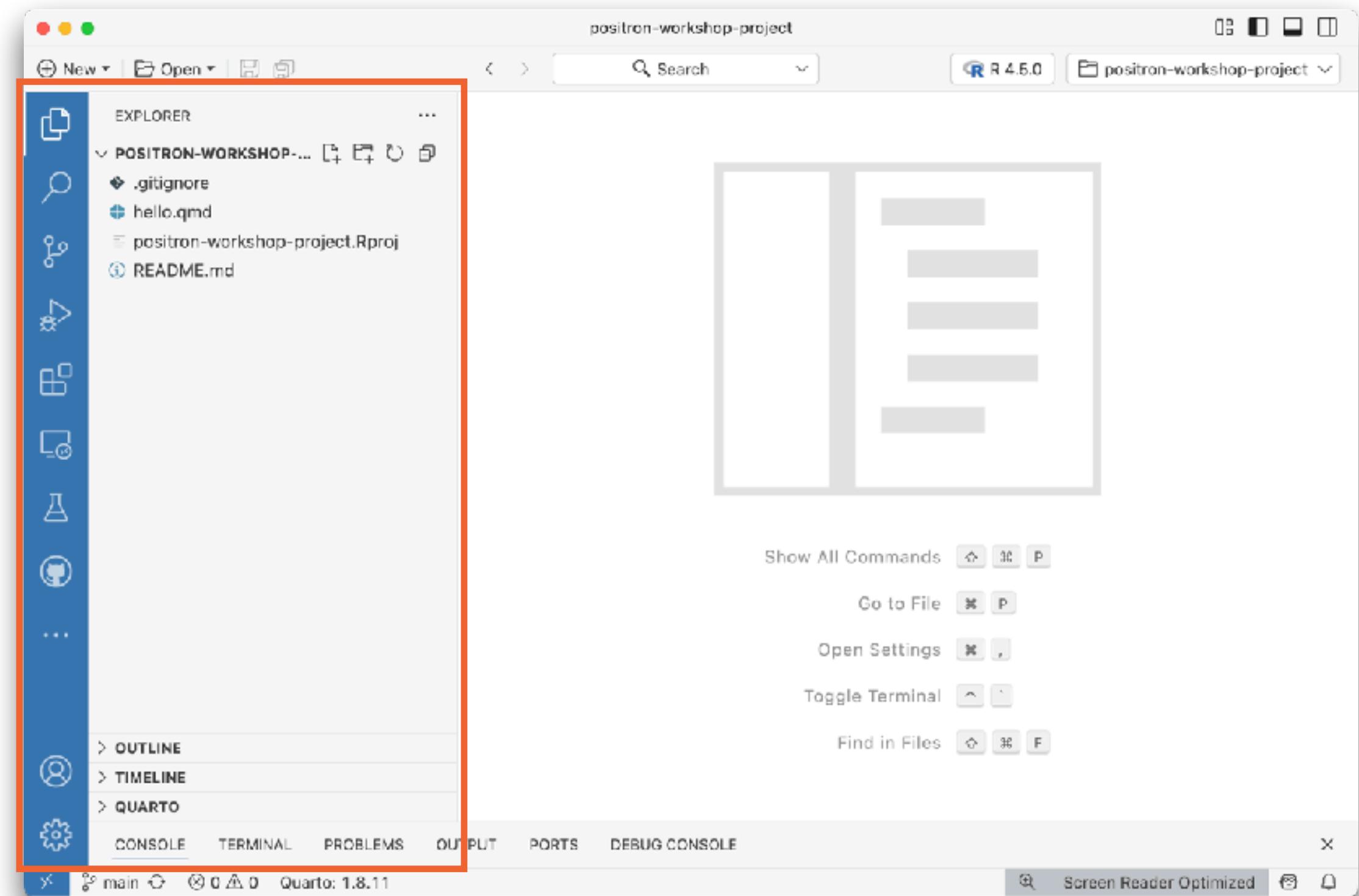
Primary side bar

- 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.

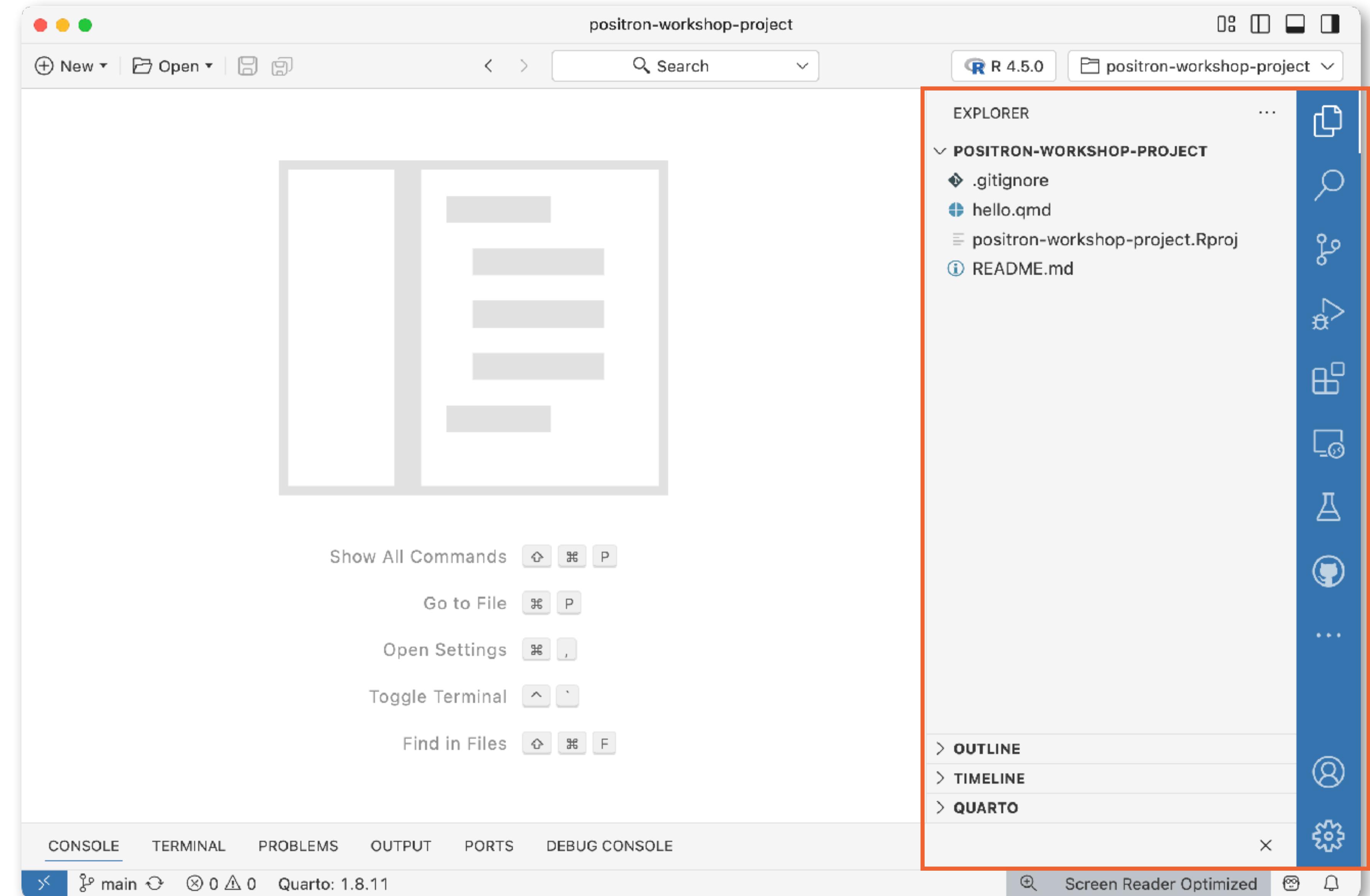
Demo

Customizing layout

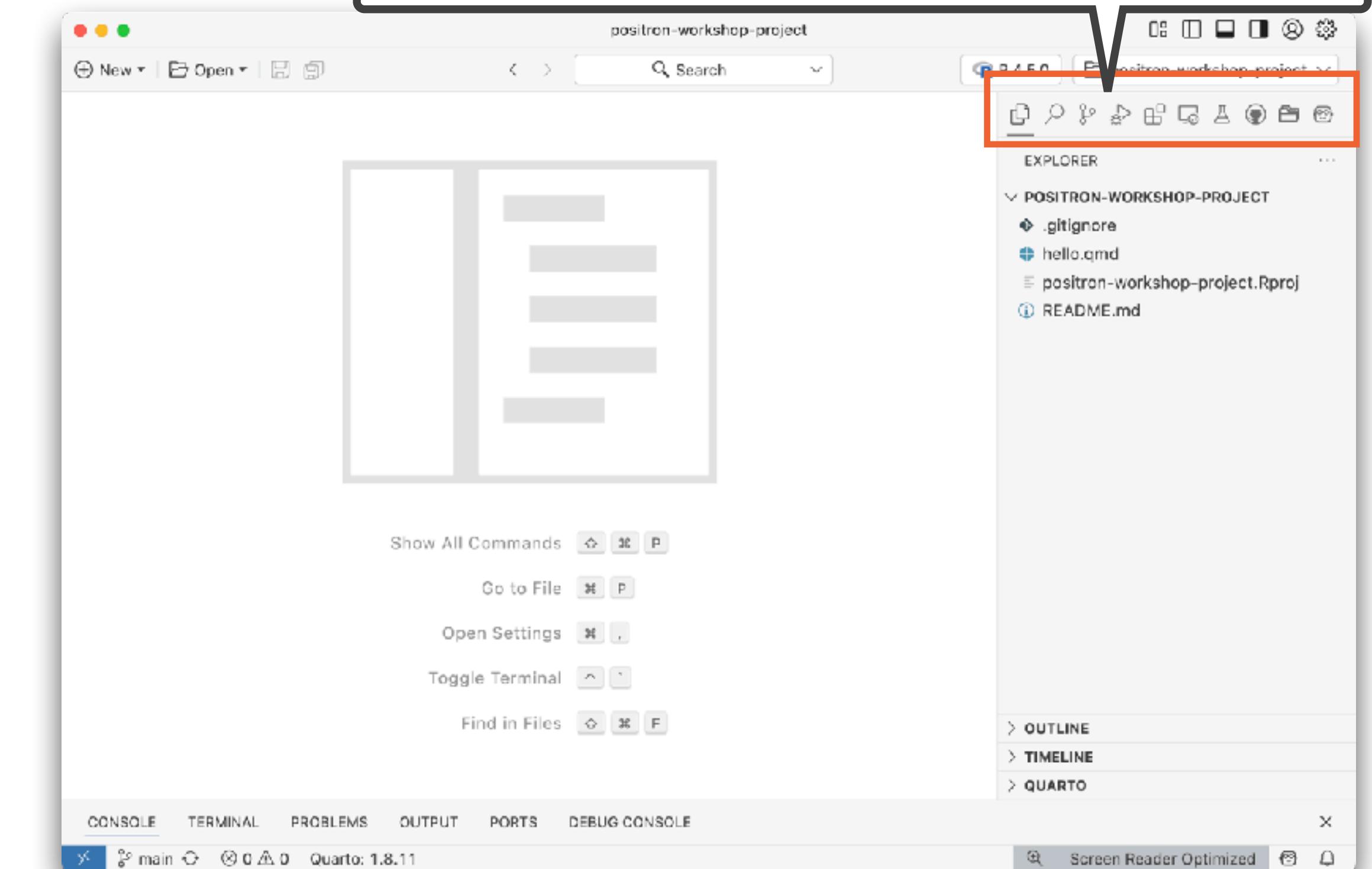
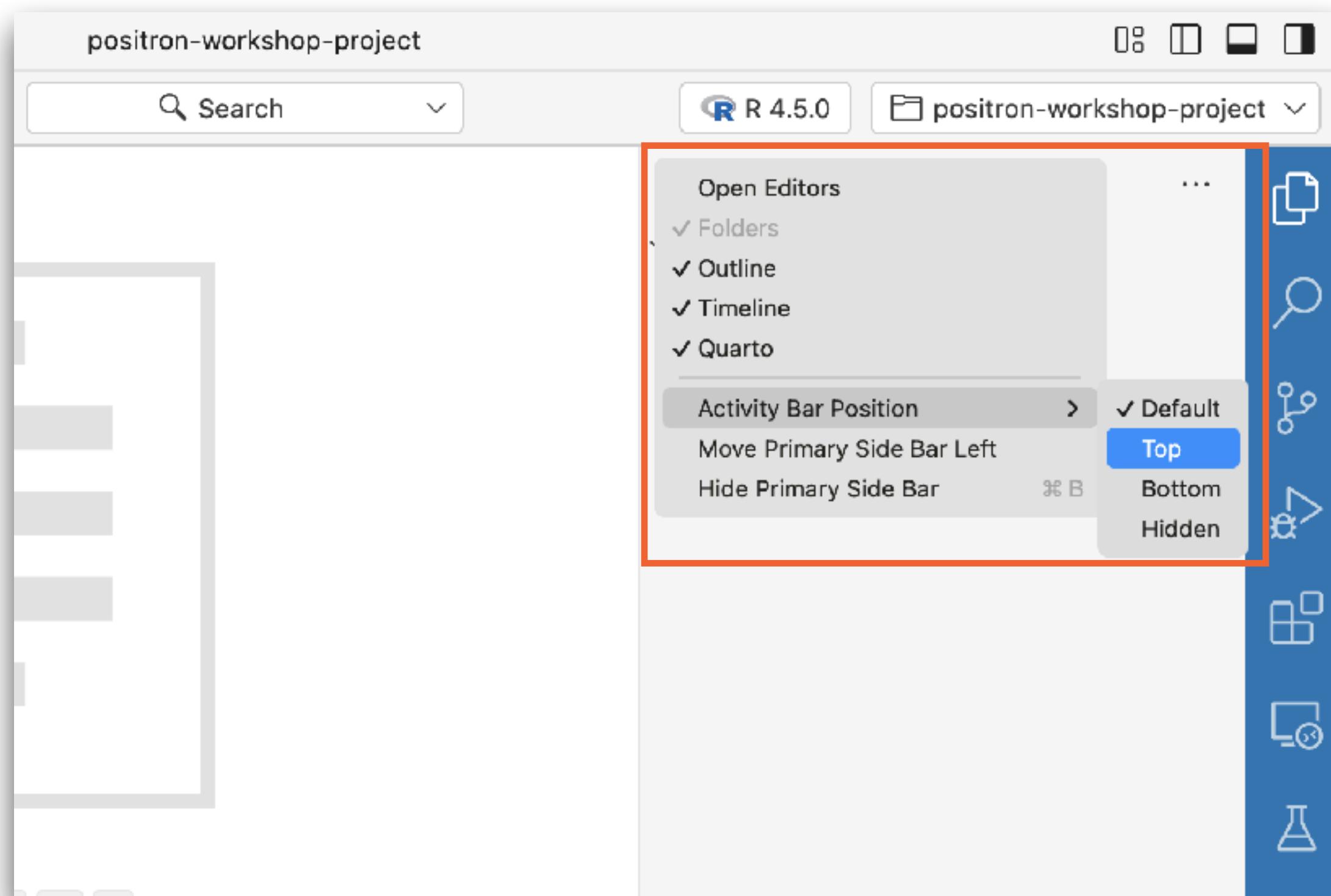
- Move the primary side bar to the right.
- Move the activity bar to the top.



Right click to
reveal menu

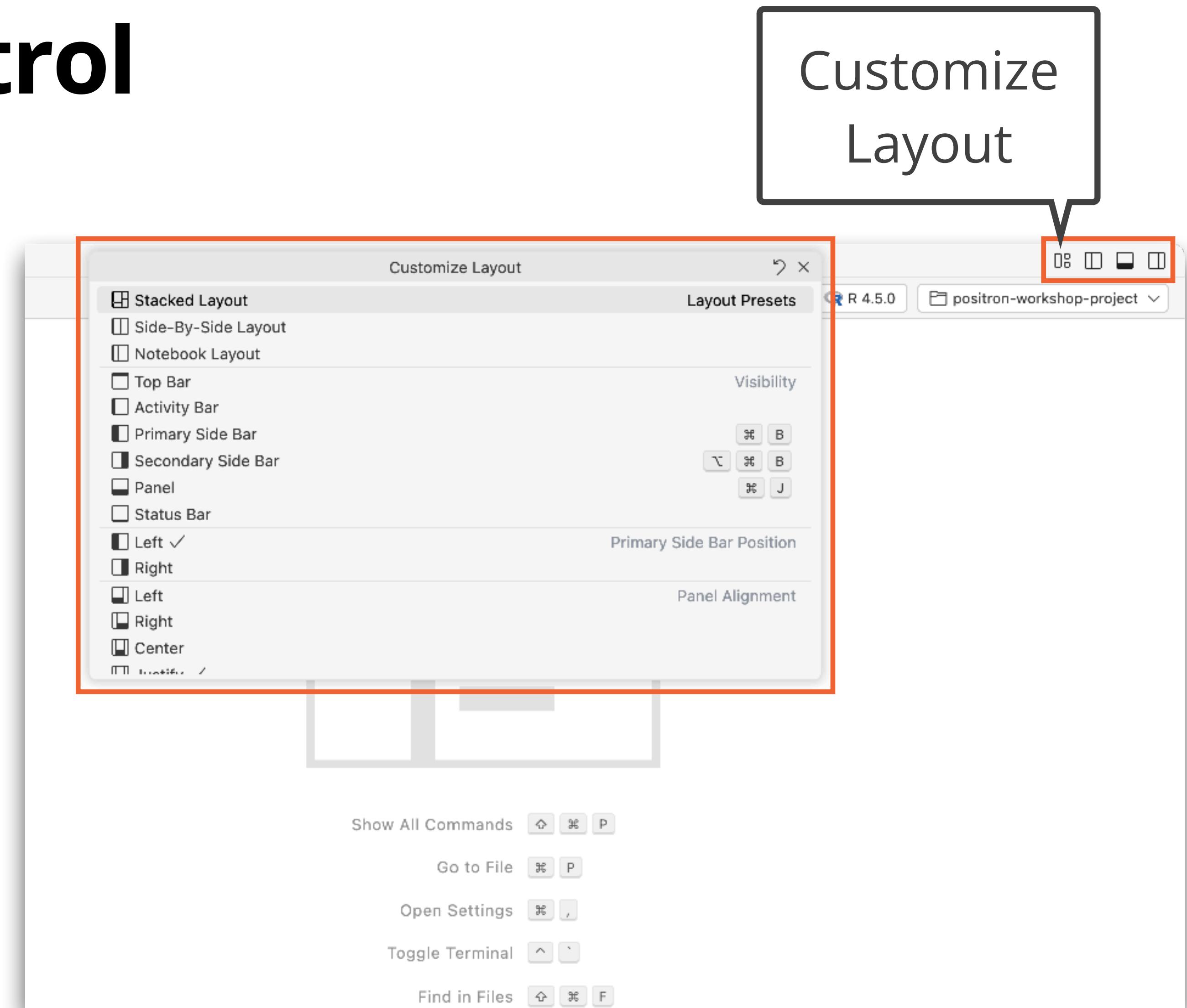


Activity Bar on top,
and not blue anymore.



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 the Quarto document editor interface. On the left, there's a preview area with a penguin illustration labeled 'CHINSTRAP!', 'GENTOO!', and 'ADELIE!'. Below the preview is a section titled 'Meet the penguins' with text about the penguins dataset from the `palmerpenguins` package. At the bottom of the preview area, there's some R code for plotting penguin data. To the right of the preview is the R console window showing R code and its output, including a warning message about missing values. Below the console is the 'VARIABLES' panel showing the `penguins` dataset. At the bottom is the 'PLOTS' panel displaying a scatter plot titled 'Flipper and bill length' with 'Dimensions for penguins at Palmer Station LTER'. The plot shows three data series: 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

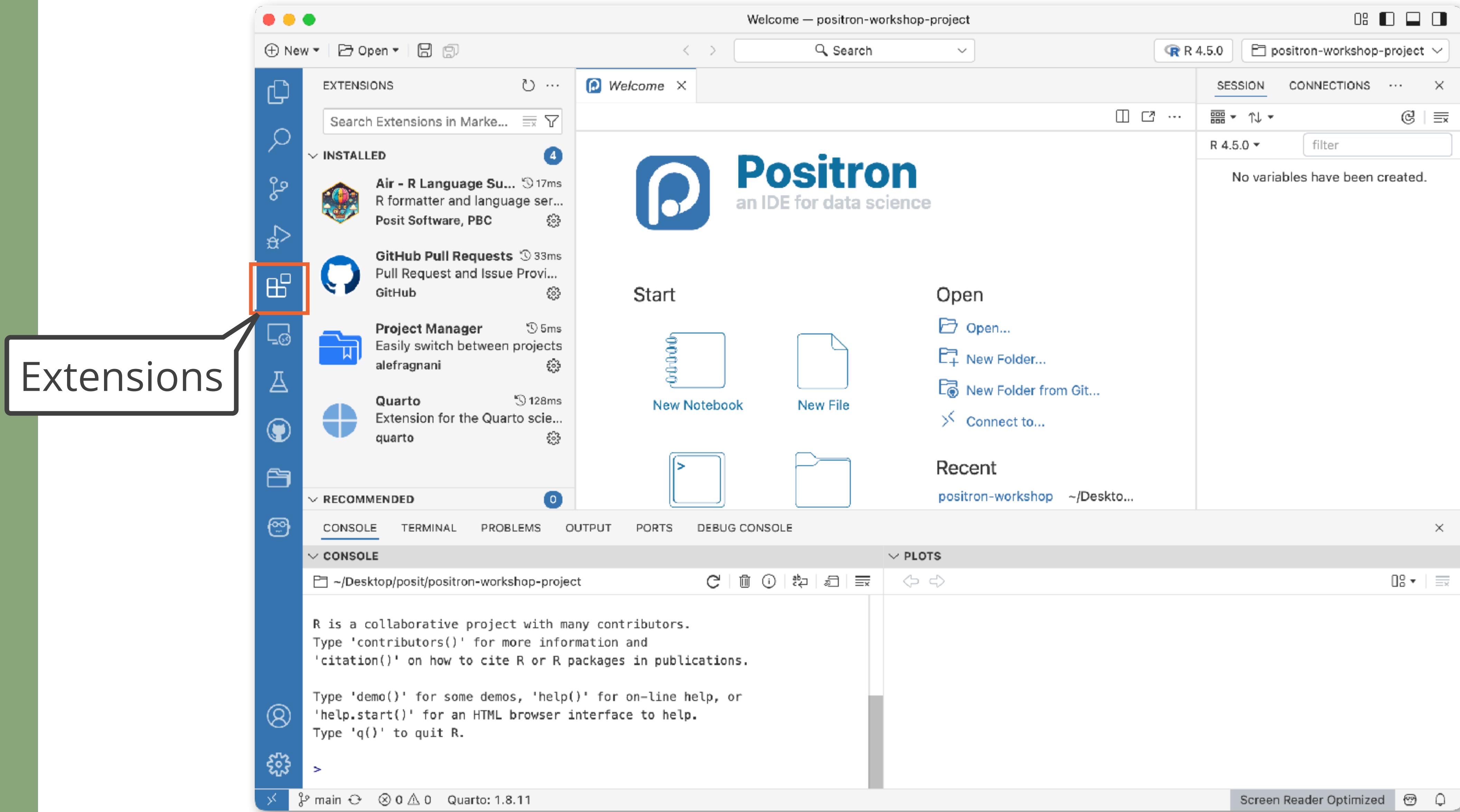
- Extensions let you add languages, debuggers, tools, themes, etc. to Positron to support your development workflow (just like in VS Code).
- Your Positron and VS Code extensions are separate and won't interfere with each other.
- Unlike in VS Code, you install Positron extensions from [Open VSX](#).

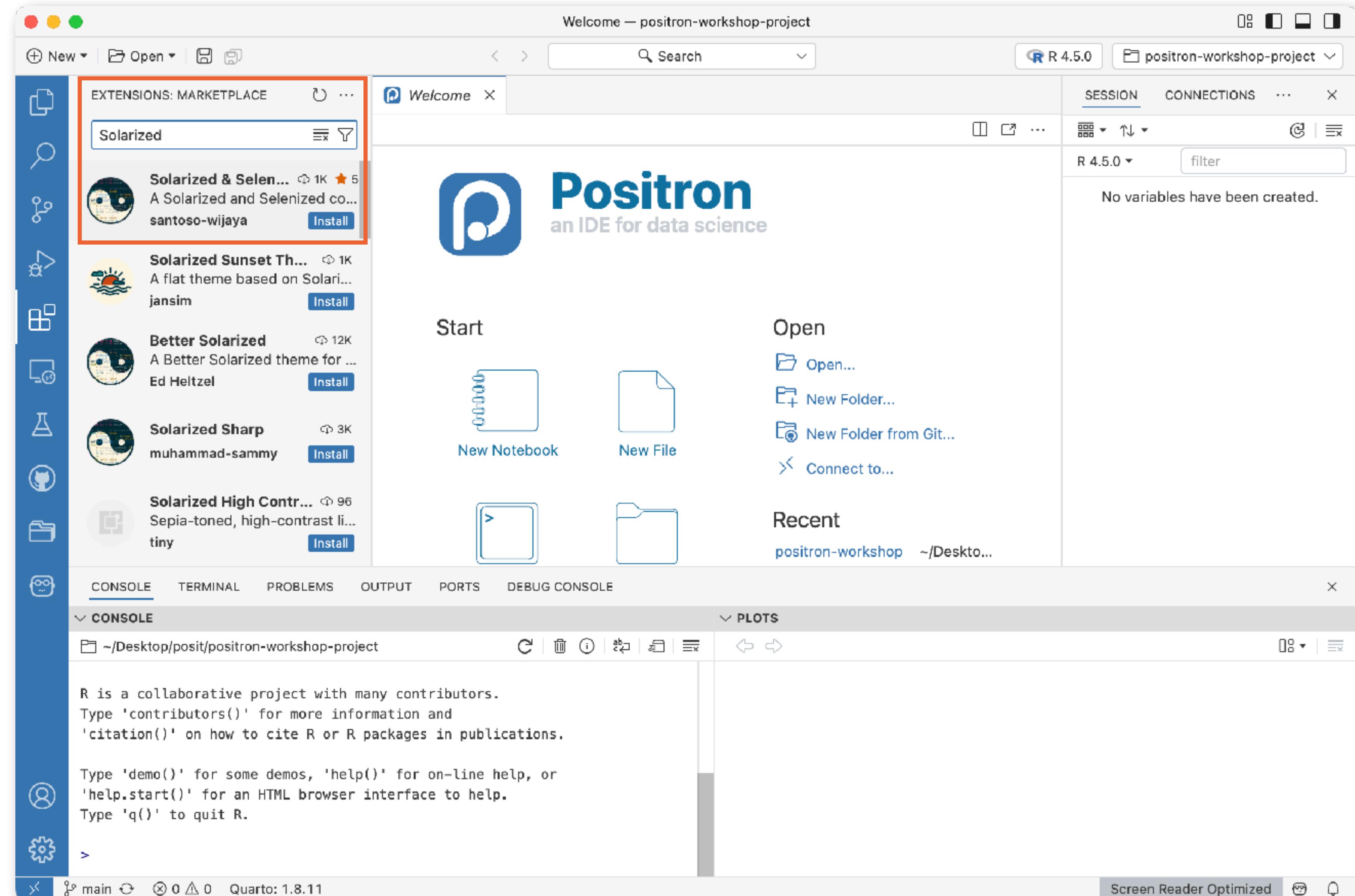
3m 00s

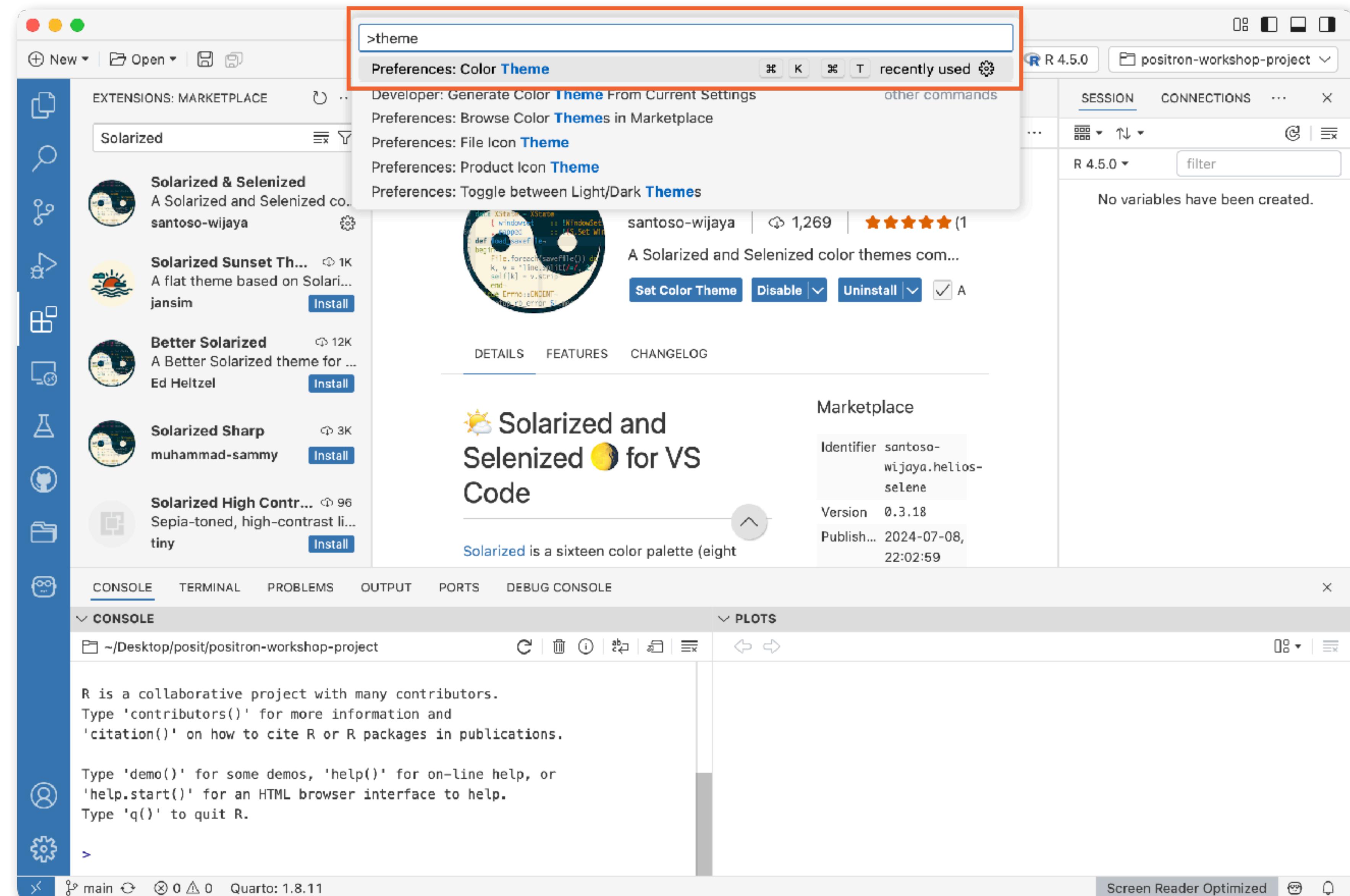
Your turn

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.







Select Color Theme (detect system color mode disabled)

+ Browse Additional Color Themes...

Positron Light Default Positron Light light themes ⚙

Selenized Light Selene Selenized Light

Solarized Light Helios Solarized Light

Positron Dark Default Positron Dark dark themes ⚙

Selenized Dark Selene Selenized Dark

Solarized Dark Helios Solarized Dark ⚙

Tomorrow Night Blue

Light High Contrast Default High Contrast Light high contrast themes ⚙

Dark High Contrast Default High Contrast

Solarized and Selenized A Solarized and Selenized color theme for VS Code santoso-wijaya ⚙

Solarized Sunset Theme A flat theme based on Solarized jansim ⚙ **Install**

Better Solarized A Better Solarized theme for VS Code Ed Heltzel ⚙ **Install**

Solarized Sharp muhammad-sammy ⚙ **Install**

Solarized High Contrast Sepia-toned, high-contrast light theme tiny ⚙ **Install**

SESSION CONNECTIONS ... X

R 4.5.0 positron-workshop-project

No variables have been created.

DETAILS FEATURES CHANGELOG

Marketplace

Identifier santoso-wijaya.helios-selene

Version 0.3.18

Publish... 2024-07-08, 22:02:59

CONSOLE TERMINAL PROBLEMS OUTPUT PORTS DEBUG CONSOLE

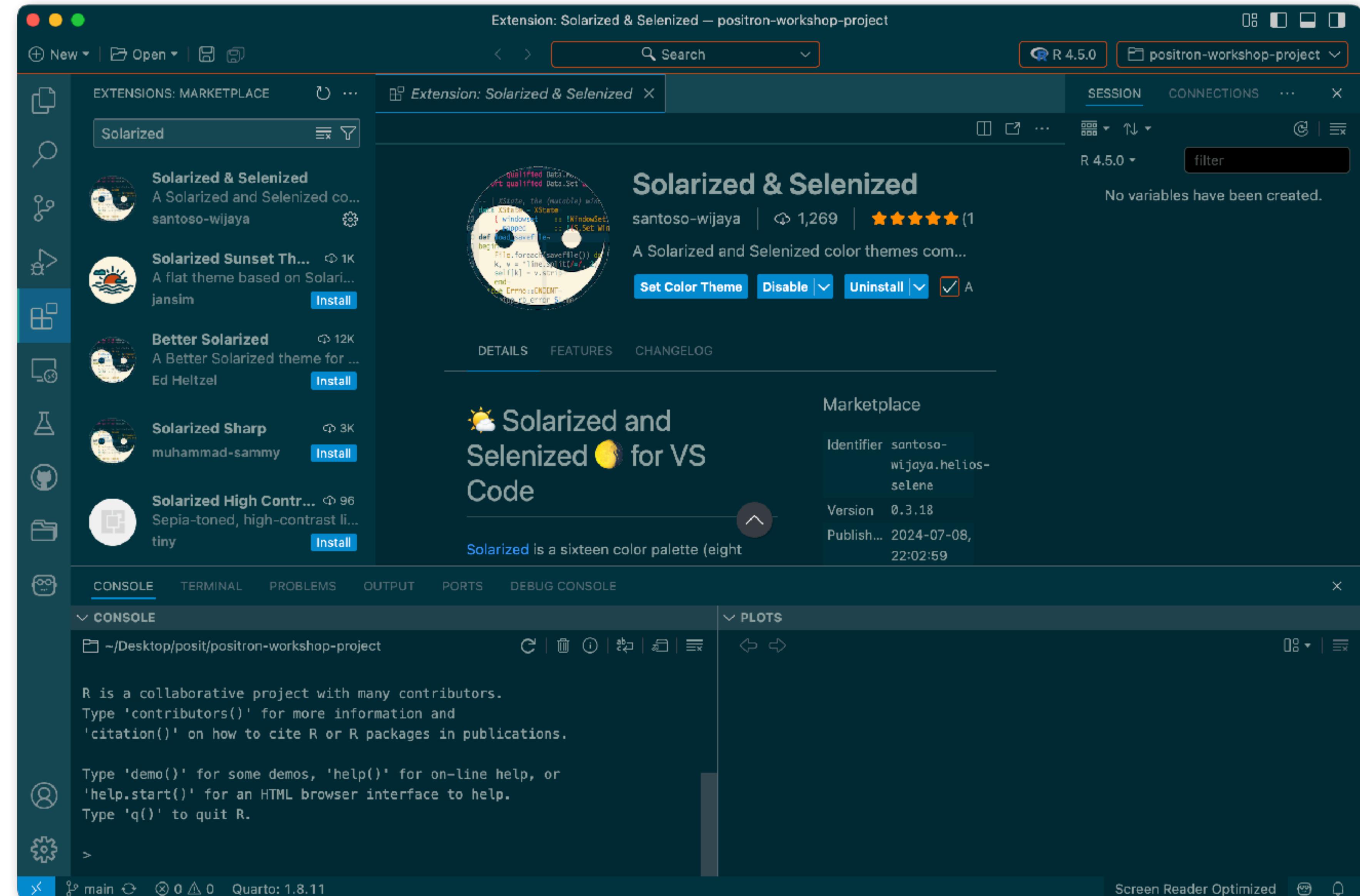
CONSOLE ~/Desktop/posit/positron-workshop-project C | ⌛ | ⓘ | ⚙ | ⌂ | ⚙

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.

>

Screen Reader Optimized ⚙ ⚙



Solarized
Dark
theme

Built-in extensions

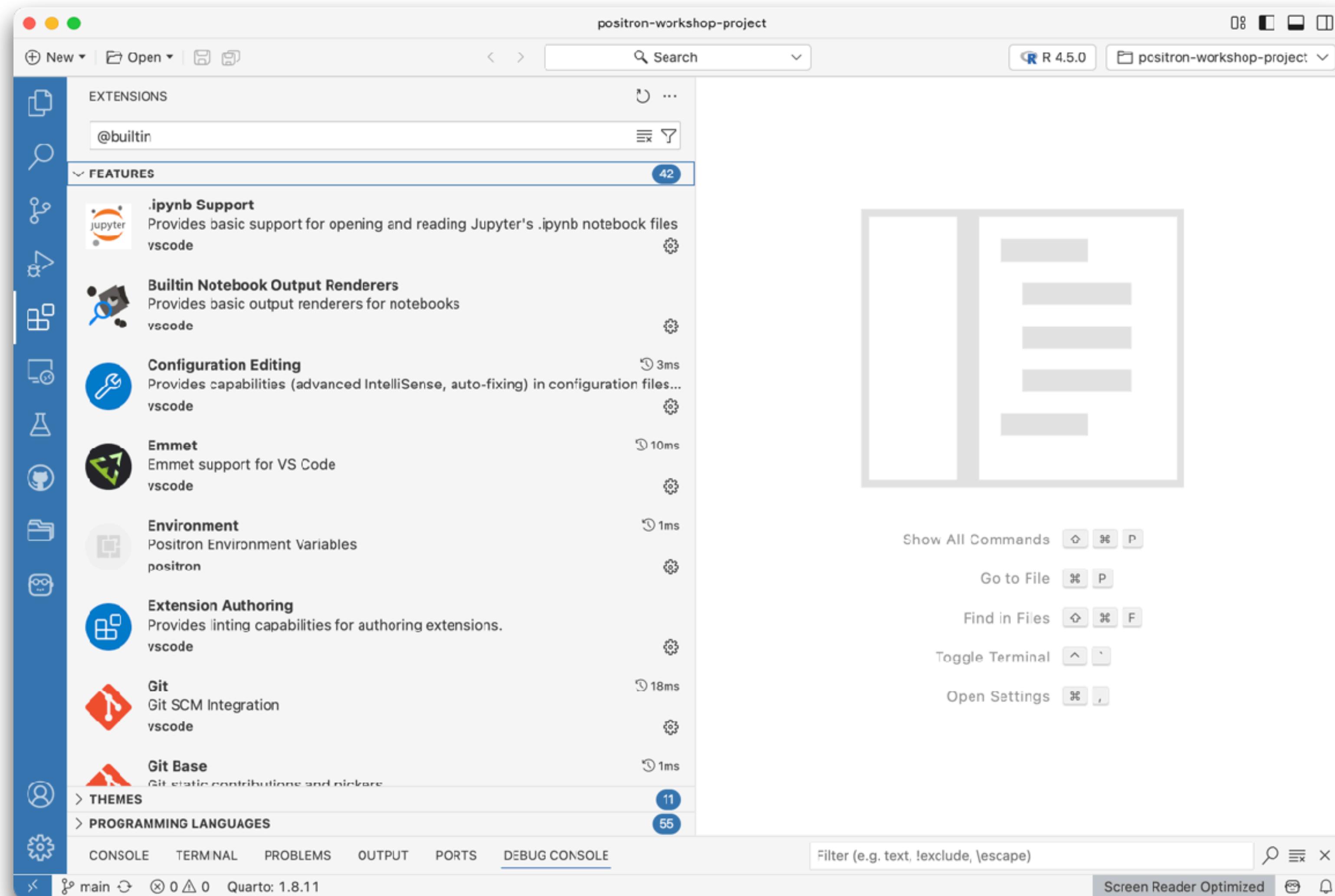
- Built-in extensions (e.g., R, Positron Python, and Markdown Language Basics) are installed with Positron.
- They can only be updated by updating Positron.
- They can be seen in the extensions view with `@builtin`.

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

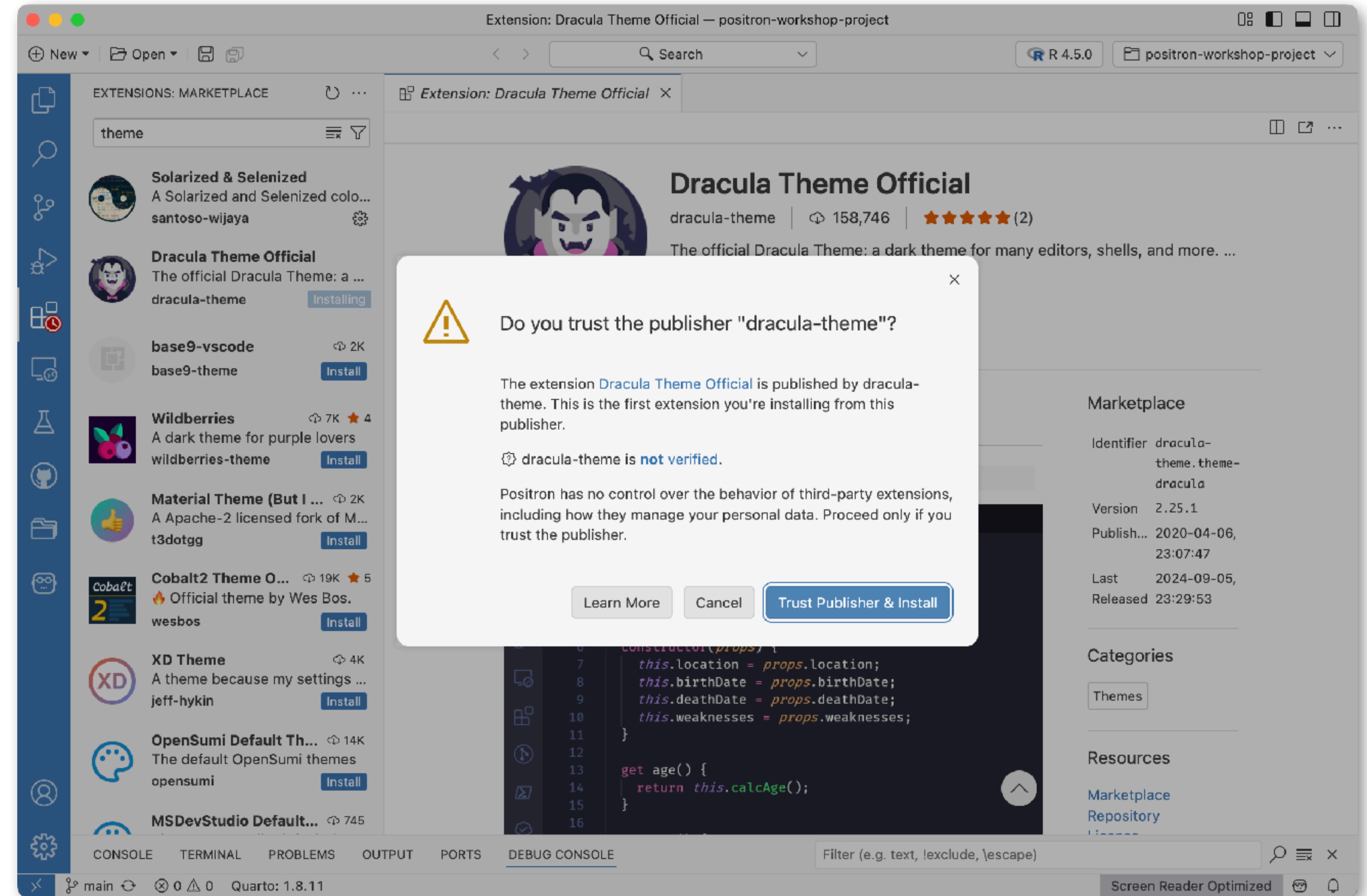
- Any other extensions need to be installed from Open VSX.
- Popular extensions you might consider installing are Project Manager, Shiny, Rainbow CSV, indent-rainbow, etc.
- You can also install Extension Packs; a collection of packs curated by users for specific tastes, tasks, or languages.

5m 00s

Your turn

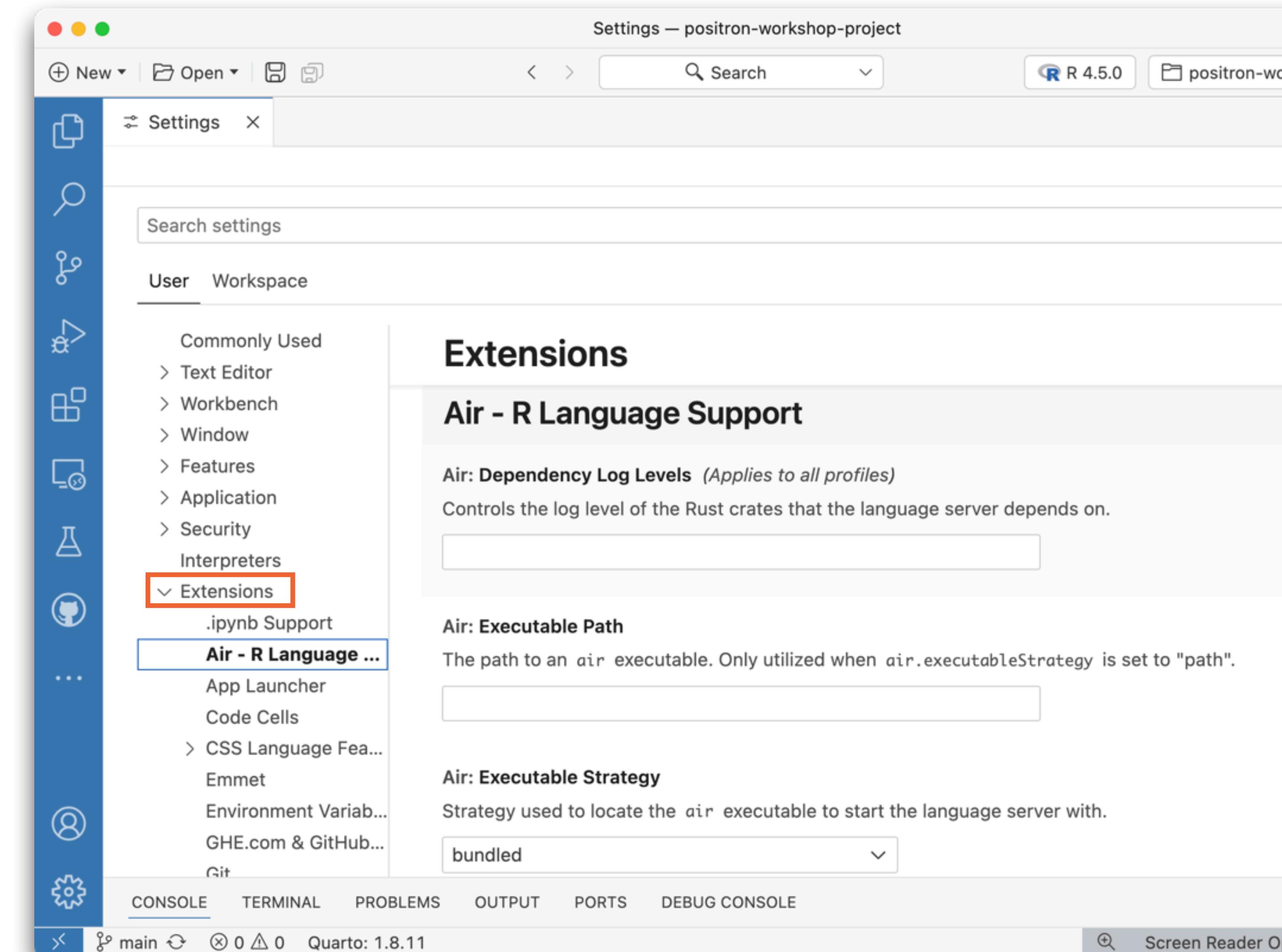
Install the ugliest theme extension

- 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!



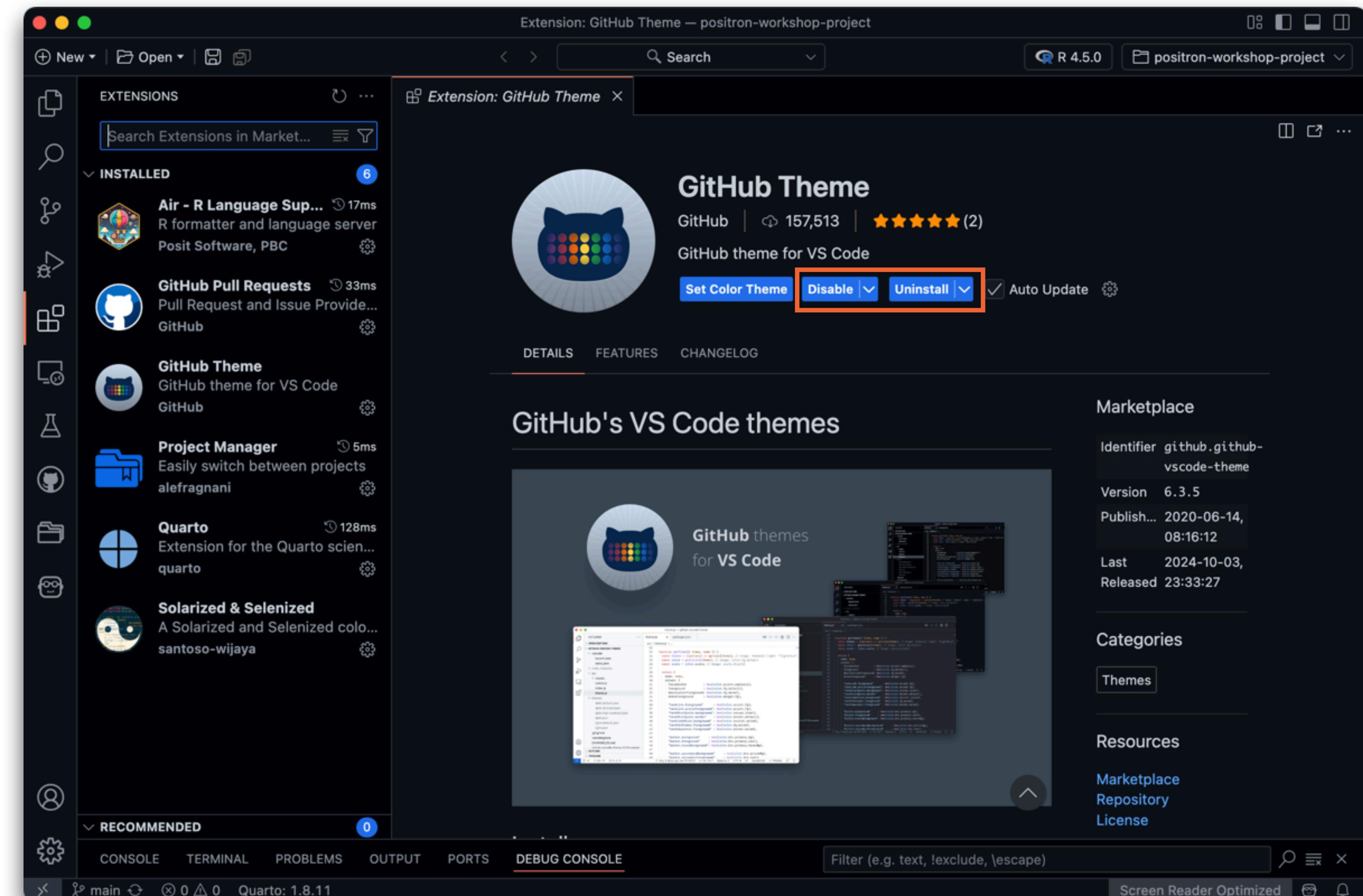
Extension settings

- 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