Setting up Positron

Daniel Padfield

July 9, 2025

Table of contents

# Outline

# Things to cover

* Downloading and installing Positron.

# External links

Smarter people than me have written about how they use Positron, so I will list a few of the ones I found most useful here.

# Neat things about Positron

* Has an automatic formatter (air) which formats R code on save.
* Can easily run both R code and bash commands depending on the script.
* Can easily SSH into remote servers to run bash code (not sure where the R code goes).
* Has nice git integration
* No longer uses RStudio projects, uses folders instead.

# Installing Positron and setting up

* Download Positron from the Positron website (https://positron.com/download).
* Check git is installed by running git --version in the terminal.
* Check quarto is installed by running quarto --version in the terminal.
* Install miniforge for installing bioinformatics packages.

# Things I have done

* Set up GitHub Copilot for autocompletion using [this](https://positron.posit.co/assistant.html).
* Set up a code snippet for starting bash and R scripts.
* Set up a few keybindings.

# Keybindings

* Ctrl/(Cmd on Mac) + Shift + P to open the command palette. This is the most important keybinding to know. From this you can search for a bunch of commands. Your most recent ones are saved at the top of the list.
* Ctrl/(Cmd on Mac) + Shift + P *Run selected text in active terminal*. This allows you to interactively script and test code in the terminal.
* Ctrl/(Cmd on Mac) + Shift + P *Change Language Mode* allow you to change the language of the current script.
* cmd+t to open and focus on the terminal.
* cmd+r to focus on the active editor group.
* cmd+e to focus on the R console.
* cmd+r to accept a code suggestion from GitHub Copilot.
* You can change Keyboard Shortcuts by opening the command palette and searching for *Preferences: Open Keyboard Shortcuts*. You can then search for a command and change its keybinding. You can also do this manually by opening the command palette and searching for *Preferences: Open Keyboard Shortcuts (JSON)*. This will open a JSON file where you can add your own keybindings.

My current keybinding file is as follows (it likely has some errors in but works for me at the moment):

// Place your key bindings in this file to override the defaults  
[  
 {  
 "key": "cmd+t",  
 "command": "-workbench.action.showAllSymbols"  
 },  
 {  
 "key": "cmd+t",  
 "command": "workbench.action.terminal.toggleTerminal",  
 "when": "terminal.active"  
 },  
 {  
 "key": "ctrl+`",  
 "command": "-workbench.action.terminal.toggleTerminal",  
 "when": "terminal.active"  
 },  
 {  
 "key": "cmd+r",  
 "command": "workbench.action.focusActiveEditorGroup"  
 },  
 {  
 "key": "cmd+e",  
 "command": "workbench.action.positronConsole.focusConsole"  
 },  
 {  
 "key": "cmd+right",  
 "command": "editor.action.inlineSuggest.commit",  
 "when": "inlineEditIsVisible && tabShouldAcceptInlineEdit && !editorHoverFocused && !editorTabMovesFocus && !suggestWidgetVisible || inlineSuggestionHasIndentationLessThanTabSize && inlineSuggestionVisible && !editorHoverFocused && !editorTabMovesFocus && !suggestWidgetVisible || inlineEditIsVisible && inlineSuggestionHasIndentationLessThanTabSize && inlineSuggestionVisible && !editorHoverFocused && !editorTabMovesFocus && !suggestWidgetVisible || inlineEditIsVisible && inlineSuggestionVisible && tabShouldAcceptInlineEdit && !editorHoverFocused && !editorTabMovesFocus && !suggestWidgetVisible"  
 },  
 {  
 "key": "tab",  
 "command": "-editor.action.inlineSuggest.commit",  
 "when": "inlineEditIsVisible && tabShouldAcceptInlineEdit && !editorHoverFocused && !editorTabMovesFocus && !suggestWidgetVisible || inlineSuggestionHasIndentationLessThanTabSize && inlineSuggestionVisible && !editorHoverFocused && !editorTabMovesFocus && !suggestWidgetVisible || inlineEditIsVisible && inlineSuggestionHasIndentationLessThanTabSize && inlineSuggestionVisible && !editorHoverFocused && !editorTabMovesFocus && !suggestWidgetVisible || inlineEditIsVisible && inlineSuggestionVisible && tabShouldAcceptInlineEdit && !editorHoverFocused && !editorTabMovesFocus && !suggestWidgetVisible"  
 },  
 {  
 "key": "cmd+right",  
 "command": "editor.action.inlineSuggest.commit",  
 "when": "inInlineEditsPreviewEditor"  
 },  
 {  
 "key": "tab",  
 "command": "-editor.action.inlineSuggest.commit",  
 "when": "inInlineEditsPreviewEditor"  
 },  
 {  
 "key": "cmd+b",  
 "command": "quarto.toggleBold"  
 },  
 {  
 "key": "cmd+i",  
 "command": "quarto.toggleItalic"  
 }  
]