Session 1: Installation Guide for R and Visual Studio Code for Screen Reader Users

JooYoung Seo

2023-08-02

Install R, Visual Studio Code, and Related Dependencies

this section describes how to install R, Visual Studio Code, and other related dependencies. The following instructions are tested on Windows 10 and Mac OS.

Windows

1. Press `Windows+R` and type "cmd" without the quotes. After typing, don't hit `Enter`. Instead, press `Ctrl+Shift+Enter` to run the command prompt as administrator privilege. If you are prompted for an administrator password or confirmation, type the password or provide confirmation.
2. Copy and paste the following command into the command prompt and press enter. This will install chocolatey, a package manager for Windows. Note that there is copy button below the command. You can click the copy button to copy the command. To paste the command into the command prompt, you can press Ctrl+V.

```
@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -
```

3. Copy and paste the following command into the opened command prompt and press enter. This will install R, VSCode and other related dependencies. Grab a cup of coffee and wait for the installation to finish. It may take a while. You will hear a "do mi sol" beep sound when the installation is completed. If it fails to install, you will hear a "do do do" beep sound. If you hear the "do do do" beep sound, you can try to run the command again.

```
powershell -Command "& { $sapi = New-Object -ComObject SAPI.SpVoice; $sapi.Rate = 10; $packa
```

4. After installation, type the following in the opened command prompt and hit enter key. This will add R to your path environment variable. You will hear a "do mi sol" beep sound when the installation is completed. If it fails to install, you will hear a "do do do" beep sound. If you hear the "do do do" beep sound, you can try to run the command again.

```
powershell -Command "& { $sapi = New-Object -ComObject SAPI.SpVoice; $sapi.Rate = 10; try { $
```

5. Close the command prompt by pressing `Alt+F4` key.
6. Press `Windows+R` and type "cmd" without the quotes. After typing, don't hit `Enter`. Instead, press `Ctrl+Shift+Enter` to run the command prompt as administrator privilege. If you are prompted for an administrator password or confirmation, type the password or provide confirmation.
7. In the opened command prompt, copy and paste the following command and press enter. This will install R packages that are required for VSCode.

```
@refreshenv && Rscript -e "packages <- c('languageserver', 'lintr', 'httpgd', 'DT', 'beepr',
```

8. The following is optional. However, I recommend you to install the latest version of PowerShell to benefit from VSCode shell integration feature. You can copy and paste the following command into the opened command prompt and press enter. Note that the default PowerShell version on Windows 10 is 5.1. The following command will install PowerShell 7 or above.

```
winget install Microsoft.PowerShell
```

9. Go to Section **??**, to complete the remaining steps.

Mac OS

The following assumes that you have installed Homebrew on your system. If you haven't installed Homebrew, you can install it by following the instruction described in the Homebrew website.

1. Open terminal by pressing `Command+Space` and type "terminal" without the quotes. Press enter to open the terminal.

2. Copy and paste the following command into the terminal and press enter. Note that there is copy button below the command. You can click the copy button to copy the command. To paste the command into the command prompt, you can press CMD+V. Some commands may require you to enter your password. If so, you will be prompted to enter your password. You will not see any characters when you type your password. Just type your password and press enter.

```
#  Install R
brew install r

# Install Pandoc
brew install pandoc

# Install Quarto
brew install quarto

# Install VSCode
brew install --cask visual-studio-code

# Install xcode command line tools
xcode-select --install

# Install TinyTeX
quarto install tinytex
```

3. In the terminal, copy and paste the following command and press enter. This will install R packages that are required for VSCode.

```
Rscript -e 'packages <- c("languageserver", "lintr", "httpgd", "DT", "beepr", "devtools"); in
```

4. Go to Section **??**, to complete the remaining steps.

Visual Studio Code Configuration

Once you have installed R and Visual Studio Code following the instruction described in Section **??**, you need to configure VSCode to make it accessible. I have created an accessible VSCode profile for data science work for your convenience. This profile contains a set of keyboard shortcuts, settings, and extensions that are useful for screen reader users when working with R and Python. You can apply this profile by following the steps below:

1. Open VSCode. You can do this by pressing `Windows+R` and type "code" without the quotes and press enter. On Mac, you can press `Command+Space` and type "code" without the quotes and press enter.
2. In VSCode, press `Ctrl+Shift+P` (on Windows) or `Command+Shift+P` (on Mac) to open the command palette.
3. Type "import profile" without the quotes and press enter.
4. If you hear "Provide Profile Template URL - Import from Profile Template..." copy and paste the following URL into the opened input box and press enter.

```
https://insiders.vscode.dev/profile/github/0cbbd65e63c766f7d37586e246d033f6
```

5. Press `Ctrl+0` (on Windows) or `Command+0` (on Mac) to open the sidebar. You will hear "data_science_accessible Tree View."
6. Press `Ctrl+LeftArrow` (on Windows) or `Command+LeftArrow` (on Mac) to collapse all the opened tree views in the sidebar.
7. Keep all the checkboxes checked and press `Tab` key multiple times until you hear "Create Profile" button and press `Enter` key.

To verify whether the profile is applied correctly, you can check the title bar of VSCode. If you hear "data science accessible - Visual Studio Code," the profile is applied correctly.

RProfile Settings

After configuring VSCode accessibility as described in Section **??**, you need to add the following code to your `Rprofile`. The `Rprofile` is a script that R runs at startup. You can use this script to customize R startup behavior to suit your personal preferences.

I have created an accessible `Rprofile` for your convenience. This `Rprofile` contains a set of options that are useful for screen reader users. To apply this `Rprofile`, folow the steps below:

1. Open VSCode. You can do this by pressing `Windows+R` and type "code" without the quotes and press enter. On Mac, you can press `Command+Space` and type "code" without the quotes and press enter.
2. In VSCode, press `Ctrl+Shift+P` (on Windows) or `Command+Shift+P` (on Mac) to open the command palette.
3. Type "create r terminal" without the quotes and press enter.

4. In the opened R terminal, type the following command and press enter. This will open your `Rprofile` in VSCode.

```
usethis::edit_r_profile()
```

5. In the opened `Rprofile`, press `Ctrl+End` (on Windows) or `Command+DownArrow` (on Mac) to move to the end of the file.

6. Copy and paste the following code to the end of the file. Please don't manually copy the code as it is too long. Instead, use the copy button below the code to copy the code. To paste the code into the `Rprofile`, you can press `Ctrl+V` (on Windows) or `Command+V` (on Mac).

```
# Useful option 1
## Setting text-based interaction instead of R GUI dialog box; especially, useful when choos
## Since R GUI dialog box is not accesible, blind R users tend to pre-define their CRAN mirr
## Furthermore, when updating R packages, users often encounter inaccessible dialog box aski
options(menu.graphics = FALSE)

# Useful option 2
## Setting pager for your preferable text editor instead of the R internal one which is inac
## For example, if you would like to know what dataset are available under "ggplot2" package
## If you set your pager option like below, you would be able to check the results in your d
## This option can be instrumental in other functions using pager option like `reticulate::p
options(pager = getOption("editor"))

# Useful option 3
## Playing beep sound per error and use {rlang} style error trace:
options(error = function() {
  rlang::entrace()
  beepr::beep()
})

# Useful option 4
## Turn off unicode for better screen reader readability
options(cli.unicode = FALSE)

# Useful option 5
## Always use RStudio CRAN mirror for downloading packages
options(repos = c(CRAN = "https://cran.rstudio.com"))

# Useful option 6
## Give more information about the LaTeX error
```

```r
options(tinytex.verbose = TRUE)

# Specific options for VSCode:

## Use VSCode as the default R editor
options(editor = "code")

## VSCode R initializer and other options
if (interactive() && Sys.getenv("RSTUDIO") == "") {
  # Load `init.R` provided by vscode-r extension: This is required to attach any user-custom
  source(file.path(Sys.getenv(if (.Platform$OS.type == "windows") "USERPROFILE" else "HOME")

  # Use an external browser for displaying html files, such as {gt} and {xaringan}
  options(vsc.viewer = FALSE)

  # Use an external browser for web apps, such as {shiny}
  options(vsc.browser = FALSE)

  # View help page in an external browser
  options(vsc.helpPanel = FALSE)

  # Use original data viewer (i.e., `utils::View()`)
  options(vsc.view = FALSE)
}

## Define code style according to {tidyverse}
options(languageserver.formatting_style = function(options) {
  styler::tidyverse_style(indent_by = 2L)
})

## Define a more accessible view function
View <- function(...) {
  DT::datatable(...)
}
```

7. Press `Ctrl+S` (on Windows) or `Command+S` (on Mac) to save the file.
8. Press `Ctrl+W` (on Windows) or `Command+W` (on Mac) to close the file.
9. Close R terminal by pressing `Ctrl+W` (on Windows) or `Command+W` (on Mac). You may be prompted to confirm whether you want to close the terminal. If so, press `Enter` key to confirm.