

2.4 Output formats

There are two types of output formats in the **rmarkdown** package: documents, and presentations. All available formats are listed below:

- `beamer_presentation`
- `context_document`
- `github_document`
- `html_document`
- `ioslides_presentation`
- `latex_document`
- `md_document`
- `odt_document`
- `pdf_document`
- `powerpoint_presentation`
- `rtf_document`
- `slidy_presentation`
- `word_document`

We will document these output formats in detail in Chapters 3 and 4. There are more output formats provided in other extension packages (starting from Chapter 5). For the output format names in the YAML metadata of an Rmd file, you need to include the package name if a format is from an extension package, e.g.,

```
output: tufte::tufte_html
```

If the format is from the **rmarkdown** package, you do not need the `rmarkdown::` prefix (although it will not hurt).

When there are multiple output formats in a document, there will be a dropdown menu behind the RStudio `knit` button that lists the output format names (Figure 2.4).

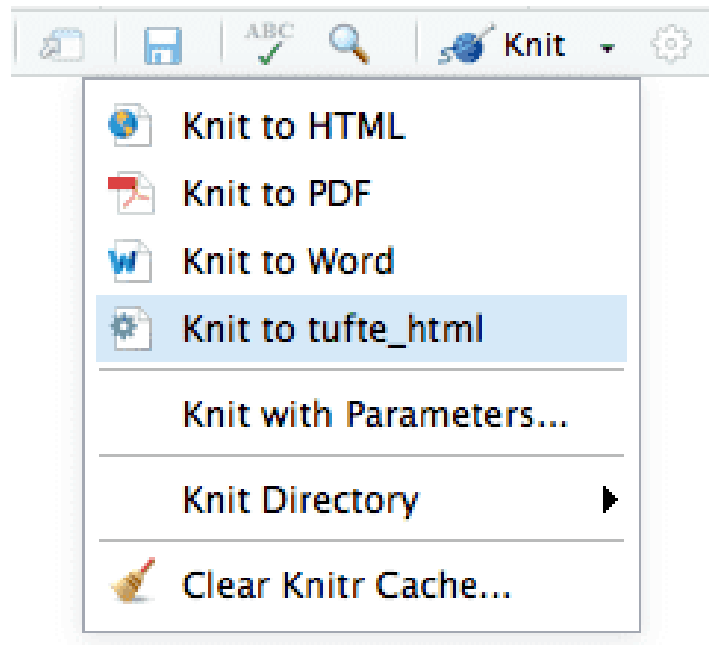


FIGURE 2.4: The output formats listed in the dropdown menu on the RStudio toolbar.

Each output format is often accompanied with several format options. All these options are documented on the R package help pages. For example, you can type `?`

`rmarkdown::html_document` in R to open the help page of the `html_document` format. When you want to use certain options, you have to translate the values from R to YAML, e.g.,

```
html_document(toc = TRUE, toc_depth = 2, dev = 'svg')
```

can be written in YAML as:

```
output:
  html_document:
    toc: true
    toc_depth: 2
    dev: 'svg'
```

The translation is often straightforward. Remember that R's `TRUE`, `FALSE`, and `NULL` are `true`, `false`, and `null`, respectively, in YAML. Character strings in YAML often do not require the quotes (e.g., `dev: 'svg'` and `dev: svg` are the same), unless they contain special characters, such as the colon `:`. If you are not sure if a string should be quoted or not, test it with the **yaml** package, e.g.,

```
cat(yaml::as.yaml(list(
  title = 'A Wonderful Day',
  subtitle = 'hygge: a quality of coziness'
)))
```

```
title: A Wonderful Day
subtitle: 'hygge: a quality of coziness'
```

Note that the subtitle in the above example is quoted because of the colon.

If you have options that need to be the result of an evaluated R expression, you can use `!expr`, which tells the **yaml** package that it needs to parse and evaluate that option. Below is an example that uses a random theme for the HTML output:

```
output:
  html_document:
    theme: !expr sample(c("yeti", "united", "lumen"), 1)
```

If a certain option has sub-options (which means the value of this option is a list in R), the sub-options need to be further indented, e.g.,

```
output:
  html_document:
    toc: true
    includes:
      in_header: header.html
      before_body: before.html
```

Some options are passed to **knitr**, such as `dev`, `fig_width`, and `fig_height`. Detailed documentation of these options can be found on the **knitr** documentation page:

<https://yihui.name/knitr/options/>. Note that the actual **knitr** option names can be different. In particular, **knitr** uses `.` in names, but **rmarkdown** uses `_`, e.g., `fig_width` in **rmarkdown**

corresponds to `fig.width` in **knitr**. We apologize for the inconsistencies—programmers often strive for consistencies in their own world, yet one standard plus one standard **often equals three standards**. If I were to design the **knitr** package again, I would definitely use `_`.

Some options are passed to Pandoc, such as `toc`, `toc_depth`, and `number_sections`. You should consult the Pandoc documentation when in doubt. R Markdown output format functions often have a `pandoc_args` argument, which should be a character vector of extra arguments to be passed to Pandoc. If you find any Pandoc features that are not represented by the output format arguments, you may use this ultimate argument, e.g.,

```
output:
```

```
pdf_document:
```

```
  toc: true
```

```
  pandoc_args: ["--wrap=none", "--top-level-division=chapter"]
```