R Markdown Is All You Need

Zhe Li

About

What Is Rmarkdown

- The rmarkdown package was developed mainly by *Allaire, JJ, Yihui Xie.*
- `R Markdown` stands on the shoulders of knitr and Pandoc.

Tasks With Rmarkdown

- Compile a single R Markdown document to a report in different formats, such as PDF, HTML, or Word.

- Create notebooks in which you can directly run code chunks interactively.

- Make slides for presentations (HTML5, LaTeX Beamer, or PowerPoint).

- Produce dashboards with flexible, interactive, and attractive layouts.

- Build interactive applications based on Shiny.

- Write journal articles.

- Author books of multiple chapters.

- Generate websites and blogs.

HTML Output

YAML Metadata

```
---
title: Habits
author: John Doe
date: March 22, 2005
output: html_document
---
```

Table of contents

```
---
title: "Habits"
output:
  html_document:
    toc: true
    toc_depth: 2
---
```

Table Of Contents

Floating TOC

- You can specify the `toc_float` option to float the table of contents to the left of the main document content.

```
---
title: "Habits"
output:
  html_document:
    toc: true
    toc_float: true
---
```

Table of Contents

Floating TOC

- `collapsed` (defaults to TRUE) controls whether the TOC appears with only the top-level (e.g., H2) headers.

- `smooth_scroll` (defaults to TRUE) controls whether page scrolls are animated when TOC items are navigated to via mouse clicks.

```
---
title: "Habits"
output:
  html_document:
    toc: true
    toc_float:
      collapsed: false
      smooth_scroll: false
---
```
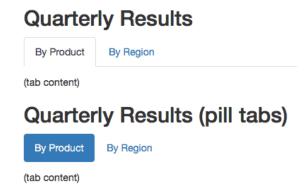
Section Numbering

```
---
title: "Habits"
output:
  html_document:
    toc: true
    number_sections: true
---
```

Tabbed sections

- You can organize content using tabs by applying the `.tabset` class attribute to headers within a document.

Quarterly Results

| By Product | By Region |

(tab content)

Quarterly Results (pill tabs)

| By Product | By Region |

(tab content)

- `.tabset-fade` attribute causes the tabs to fade in and out when switching between tabs.

- `.tabset-pills` attribute causes the visual appearance of the tabs to be "pill".

Appearance and Style

- `theme`: specifies the theme to use for the page, valid themes include default, bootstrap, cerulean, cosmo, darkly, flatly, journal, lumen, paper, readable, sandstone, simplex, spacelab, united, and yeti.

- `highlight`: specifies the syntax highlighting style including default, tango, pygments, kate, monochrome, espresso, zenburn, haddock, breezedark, and textmate.

- `smart`: indicates whether to produce typographically correct output, converting straight quotes to curly quotes. Note that smart is enabled by default.

```
---
title: "Habits"
output:
  html_document:
    theme: united
    highlight: tango
---
```

Appearance and Style

Custom CSS

- You can add your own CSS to an HTML document using the `css` option:

```
---
title: "Habits"
output:
  html_document:
    theme: null
    highlight: null
    css: styles.css
---
```

Code Folding

- **The code_folding**: hide option enables you to include R code but have it hidden by default. Users can then choose to show hidden R code chunks either individually or document wide.

```
---
title: "Habits"
output:
  html_document:
    code_folding: hide
---
```

R or Python or ???

- A less well-known fact about R Markdown is that many other languages are also supported, such as Python, Julia, C++, and SQL

```
library(reticulate)
names(knitr::knit_engines$get())
```

```
##  [1] "awk"      "bash"     "coffee"    "gawk"      "groovy"   "haskell"
##  [7] "lein"     "mysql"    "node"      "octave"    "perl"     "psql"
## [13] "Rscript"  "ruby"     "sas"       "scala"     "sed"      "sh"
## [19] "stata"    "zsh"      "highlight" "Rcpp"      "tikz"     "dot"
## [25] "c"        "cc"       "fortran"   "fortran95" "asy"      "cat"
## [31] "asis"     "stan"     "block"     "block2"    "js"       "css"
## [37] "sql"      "go"       "python"    "julia"     "sass"     "scss"
## [43] "R"        "bslib"    "targets"
```

Python

- The python engine is based on the `reticulate` package

```python
x = 'hello, python world!'
print(x.split(' '))
```

```
x = 18
print(x)
```

```
## 18
```

```
py$x
```

```
## [1] 18
```

```
py$y = 1:5
```

```
print(y)
```

```
## [1, 2, 3, 4, 5]
```

Thanks !