

統計学 2

R マークダウンの使い方

矢内 勇生

2018-04-19 (改訂: 2019-04-19, 2019-06-03, 2020-08-18, 2021-04-16, 2021-04-21)

まず、マークダウン全体のオプション（グローバルチャンクオプション）を指定する。PDF に出力する（LaTeX 使う）場合は、グラフィックデバイスに `cairo` を利用するが、HTML の場合には `cairo` を使いたくないので、`knitr::is_latex_output()` を利用して場合分けする。以下のコードチャンクは、オプションに `include = FALSE` がついているので、出力された PDF や HTML では非表示になる。このチャンク（グローバルオプション）は常に **YAML ヘッダのすぐ下** に書くべきである。

1 準備

R マークダウン (R Markdown、拡張子は `.Rmd`) ファイルは RStudio で編集する。編集したファイルを PDF（または HTML）に出力するために、`rmarkdown::render()` や `knitr::knit()` を利用する。これらがインストール済みでない場合はまずインストールする。

```
install.packages("tidyverse", dependencies = TRUE)
install.packages("rmarkdown", dependencies = TRUE)
install.packages("knitr", dependencies = TRUE)
install.packages("systemfonts")
install.packages("remotes")
remotes::install_github("Gedevan-Aleksizde/fontregisterer",
                        repos = NULL, type = "source")
```

すべてインストールできたら、パッケージを読み込む。パッケージのインストールは一度すれば済むが、`library()` を使ったパッケージの読み込みは、R を起動するたびに行う必要がある。ただし、**rmarkdown** パッケージと **knitr** パッケージは必要なときのみ呼び出せば良いので、`library()` では読み込まなくても良い。

```
library(tidyverse)
```

```
## 図のなかで日本語を使えるようにする
## フォントの設定はお好みで
```

```
## (Unix/Linux では IPAex フォントのインストールが必要かも)
if (.Platform$OS.type == "windows") {
  # Windows
  require(fontregisterer)
  my_font <- "Yu Gothic"
} else if (capabilities("aqua")) {
  # macOS
  my_font <- "HiraginoSans-W3"
} else {
  # Unix/Linux
  my_font <- "IPAexGothic"
}

## ggplot2 用の設定
theme_set(theme_gray(base_size = 9,
                      base_family = my_font))
```

2 R マークダウンの書き方・使い方

マークダウンファイル (r-markdown.Rmd) とそのファイルを元に生成された PDF ファイル (PDF を開いているならそのファイル) (r-markdown.pdf) と html ファイル (インターネットブラウザで読んでいるならそのファイル) (r-markdown.html) を見比べながら、RStudio で R マークダウンファイルを扱えるようにするのが今日の目標である。

このマークダウンをそのまま使うためには、担当教員が作ったスタイルシート (my-markdown.css) をプロジェクトのフォルダに保存する必要がある。スタイル (表示されるページの見た目) をカスタマイズしたいときは、このファイルを変更すればよい。デフォルトのスタイルのままで良いとき (あまり良くないと思うが) は、ヘッダの 'css' オプションの指定をやめる (この Rmd ファイル [html ではない] の YAML ヘッダ部分にある `css: my-markdown.css` の行を削除する)。

この資料と同じ画像を表示するためには、画像ファイルも同じフォルダに保存する必要がある。自分でもっている画像 (写真) に置き換えて試してほしい。

2.1 マークダウン記法を利用した文章の書き方

文章は、いつもどおり書けばよい。文章の一部をイタリック (斜字体) にしたいときは、イタリックにした部分を `*` または `_` で挟むと、*this is italic* あるいは *this is also italic* となる (日本語は斜字体にしない)。太字は、`**` (`*` を 2 つ) または `__` (`_` を 2 つ) で挟むと、**ここが太字** または **ここも太字** となる。太字のイタリックは、`***` (`*` を 3 つ) または `___` (`_` を 3 つ) で挟むと、***here is bold italic*** または ***here is also bold italic*** となる。

改行するときは、文章の間を 1 行以上空ける。

箇条書きは、* または _ を利用し、

- 項目 1
- 項目 2
 - 項目 2-1
 - 項目 2-2

あるいは、

- 項目 1
 - 項目 1-1
 - 項目 1-2
- 項目 2

のようにできる。* や - の後には半角スペースを挿入する。箇条書きを入れ子にするとき、字下げは Tab で行う

番号付きの箇条書きは、数字で作れる。

1. First item
2. Second item
 1. What?
 2. How?
3. Third item

のようにする。入力する際にすべて「1.」にしても、自動的に 1, 2, 3, ... という番号が降られる。全部 1 にしておくと、後から項目の並べ替えができるので楽である。(1 でなくても並べ替えるが、数字の意味が不明になる。)

ヘディング (heading) は、“#” (ハッシュ記号) で作れる。論文・レポートを書くときは、ヘディングを利用して文章を構造化する。# の数が少ないほど、上位のヘディングになる。

例を以下に示す。

3 ハッシュ 1 つのヘディング

3.1 ハッシュ 2 つのヘディング

3.1.1 ハッシュ 3 つのヘディング

■3.1.1.1 ハッシュ 4 つのヘディング 論文・レポートでは、以下のようにヘディングを使い分ける。

- #: 節 (セクション, section)

- ##: 小節 (subsection)
- ###: 小節以下の見出し (subsubsection)
- ####: (あまり使わない)

また、リンクを貼ることもできる：矢内のウェブサイト。



画像も貼れる：

(出典：いらすとや)

画像へのパスが通っていないと表示されないので注意。

HTML に出力する場合は、HTML を直接書くこともできる（PDF に出力する場合はうまくいかない）。

- リンクを貼る：矢内のウェブサイト
- 画像を貼る：

3.1.2 * 数式の書き方

LaTeX と同じように数式を書くこともできる。文章中と同じ行に数式を書きたいときは、\$で挟む。たとえば、 $\bar{x} = \sum_{i=1}^n x_i/n$ と、する。

数式を独立したブロックとして書きたいときは、\$\$で挟み、

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$$

のようにする。

3.2 コードチャンクの書き方

R のコードは、コードチャンクと呼ばれる部分に書き込む。コードチャンクは、たとえば以下のように書ける。

```
a <- 1:10
b <- -1:-10
```

R コードチャンクの始めには、3つの「`{`」の後に`{r}`をつける。`r`とスペースの後（`{}`の中）には、チャンクの名前を付ける。好きな名前を付けてよいが、他のチャンクとまったく同じ名前は付けられない。チャンクの終わりには3つの「`}`」を書く。RStudio で開いた R マークダウンファイル内でコードチャンクを作るには、次のショートカットキーを使った方がよい。

- macOS: Cmd + Option + I
- Windows: Ctrl + Alt + I

I は Insert（チャンクを「挿入する」）の頭文字である。

文章中に R コードを書きたいときは `mean(x)` のように、R のコマンドを「`'`」の間に書く。関数を実行（評価, evaluate）した後の結果を文章に入れたいときは、「`a` の平均値は `' r mean(a) '` です」のように「`r`」を入れて書くと、「`a` の平均値は 5.5 です」となる。つまり、`mean(a)` を R が計算し、その結果を文章の中に入れてくれる。この方法を使えば、文章と別に R のコマンドを実行しなくても、R の実行結果を表示することができる。

図を含めた文章も作れる。チャンクオプション `fig.cap` で図のキャプションが指定できる。

```
p <- tibble(x = rnorm(100, mean = 0, sd = 1)) %>%
  ggplot(aes(x = x)) +
  geom_histogram(binwidth = 1, color = "black", fill = "tomato") +
  labs(y = "度数", title = "ヒストグラム")
plot(p)
```

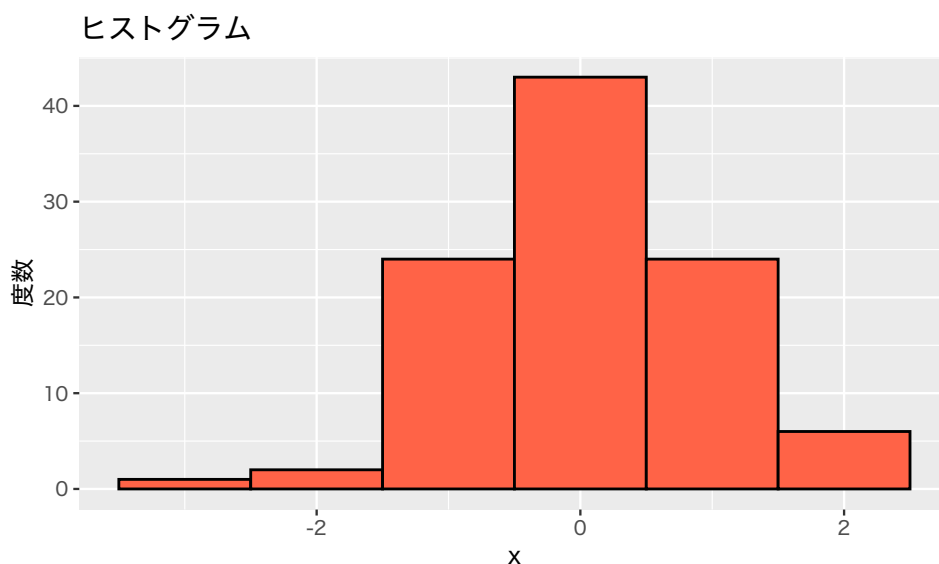


図 1: トマト色のヒストグラム

何もオプションを指定しない状態では、チャンクは1行ずつ評価され、結果も順番に次々出力される。たとえば、

```
sd(a)
```

```
## [1] 3.02765
```

```
var(a)
```

```
## [1] 9.166667
```

チャンクの最後まで評価してからまとめて結果表示したいときは、チャンクオプション **results** を 'hold' にする。オプションは、チャンク名の後に「, (comma)」を打ち、その後に書く。例えば、`{r example-option, results = 'hold'}` とチャンクの冒頭に書くと、

```
sd(a)
```

```
var(a)
```

```
## [1] 3.02765
```

```
## [1] 9.166667
```

となる。

チャンクオプションについてより詳しくはココなどを参照されたい。

また、R マークダウン全般（特に、RStudio を使う場合）については、ココを参照。

R マークダウンのチートシート（[PDF ファイル]https://github.com/rstudio/cheatsheets/blob/master/translations/japanese/Rmarkdown-cheatsheet-2.0_ja.pdf）

4 R Markdown を knit してレポートを作る

4.1 R Markdown ファイルを PDF ファイルに出力する

PDF ファイルを作るためには TeX が必要である。TeX を使ったことがなく、パソコンに TeX がインストールされていない場合は、以下のコマンドを実行して **tinytex** をインストールする。それなりに時間がかかるので気長に待とう。（注意：既に LaTeX 環境が設定済みなら、以下のコードを実行する必要はない。情報演習室のパソコンには Tex Live 2020 がインストールされているので、次のコードは実行しなくてよい）。

```
install.packages("tinytex")
tinytex::install_tinytex()
```

PDF ファイルに変換する際のオプションは、ヘッダ部分（YAML ヘッダ）で指定する。この Rmd ファイル（HTML ファイルではない。つまり、ウェブブラウザで見ている場合には表示されていない）では、第1行から第28行までがヘッダであり、そのうち、`pdf_document:` のブロックと、`documentclass:`、`classoption:` で PDF 出力のためのオプションが指定されている。例えば、`toc: true` は目次（table of contents; toc）を

表示するという指定である。非表示にするには `toc: false` とする。

試しに、“r-markdown.Rmd”を“r-markdown.pdf”に変換してみよう。Rmd ファイルを RStudio で編集している場合、コード編集画面の上にある“Knit”ボタン（毛糸と棒針のマーク）の右にある三角ボタンを押して、表示されたメニューから“Knit to PDF”を選べば PDF ができる。初めて実行するときは、足りないパッケージを自動でインストールするので、時間がかかるかもしれない。

出力された PDF ファイルは（他のディレクトリを指定しない限り）現在の作業ディレクトリ（プロジェクトのフォルダ）に保存される。出来上がった PDF ファイルを Adobe Reader や skim 等の PDF リーダで開いて確認してみよう。（RStudio の Viewr だと、平仮名の「う」が表示されないが、Adobe Reader で開くと正しく表示される。）

コマンドを使っ PDF ファイルを作るときは、`rmarkdown::render()` を使う。

```
rmarkdown::render("r-markdown.Rmd",  
                  output_format = "pdf_document",  
                  output_file = "r-markdown.pdf",  
                  run_pandoc = FALSE)
```

`run_pandoc = FALSE` を指定しないと日本語が文字化けするので注意が必要である（ボタンを押して変換するときは心配しなくてよい。ボタンで変換できればよい）。

4.2 R Markdown ファイルを HTML ファイルに出力する

HTML ファイルに変換する際のオプションは、ヘッダ部分で指定する。この Rmd ファイル（HTML ファイルではない。つまり、ウェブブラウザで見ている場合には表示されていない）では、第 1 行から第 28 行までがヘッダであり、そのうち、`html_document:` のブロックで HTML 出力のためのオプションが指定されている。例えば、`toc: true` は目次 (table of contents; toc) を表示するという指定である。非表示にするには `toc: false` とする。

試しに、“r-markdown.Rmd”を“r-markdown.html”に変換してみよう。Rmd ファイルを RStudio で編集している場合、コード編集画面の上にある“Knit”ボタン（毛糸と棒針のマーク）の右にある三角ボタンを押して、表示されたメニューから“Knit to HTML”を選べば HTML ファイルができる。

出力された HTML ファイルは（他のディレクトリを指定しない限り）現在の作業ディレクトリ（プロジェクトのフォルダ）に保存される。出来上がった HTML ファイルをウェブブラウザで開いて確認してみよう。

コマンドを使って HTML ファイルを作るときは、`rmarkdown::render()` を使う。

```
rmarkdown::render("r-markdown.Rmd",  
                  output_format = "html_document",  
                  output_file = "r-markdown.html",  
                  run_pandoc = FALSE)
```

`run_pandoc = FALSE` を指定しないと日本語が文字化けするので注意が必要である（ボタンを押して変換するときは心配なくてよい。ボタンで変換できればよい）。

[授業の内容に戻る](#)