# 計量経済学

R マークダウンの使い方

矢内　勇生

*2018-04-19*（改訂：*2019-04-19*）

## 目次



このページの内容を初めて読むときは、∗ が付いている項目は飛ばして良い（少し上級者向けの内容）。

## 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 HTML（または PDF）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)
```

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すむが、`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。

```
library("tidyverse")
library("rmarkdown")
library("knitr")
```

**Mac ユーザのみ**次のコードも実行する。

```
theme_set(theme_gray(base_size = 12, base_family = "HiraginoSans-W3"))
```

# 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 html ファイル（インターネットブラウザで読んでいるこのファイル）(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いいとき（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] のヘッダ部分にある "css: my-markdown.css" の行を削除する）。

## マークダウン記法を利用した文書の書き方

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、イタリックにしたい部分を* 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。

ヘディング (heading) は、“#”（ハッシュ記号）で作れる。# の数が少ないほど、上位のヘディングになる。

# ハッシュ 1 つのヘディング

## ハッシュ 2 つのヘディング

### ハッシュ 3 つのヘディング

#### ■ハッシュ 4 つのヘディング

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）

∗ 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。

## 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。

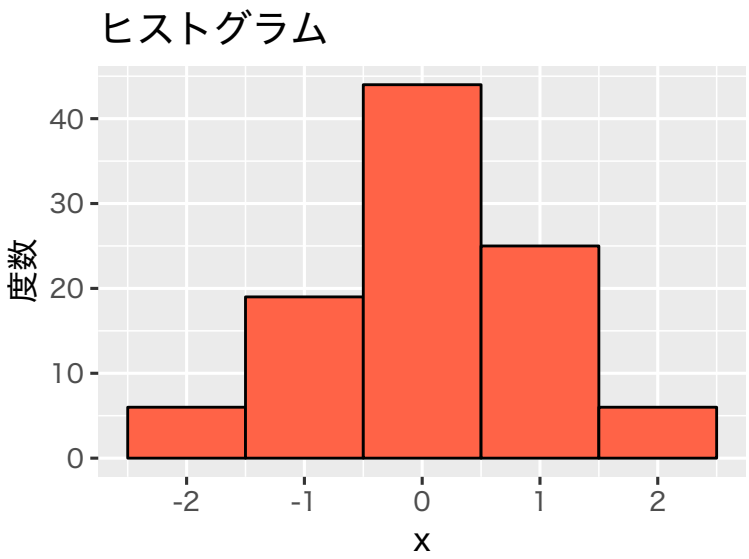
```
a <- 1:10
b <- -1:-10
```

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、ショートカットキーを使った方がよい。Ctrl + Alt + i で（3 つのキーを同時に押すと）、コードチャンクが挿入される（Mac の場合は Ctrl の代わりに Cmd）。

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中に入れたいときは、「$a$ 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。

```
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")
print(p)
```




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）もあるので、ダウンロードして持っておくと便利である。R と RStudio に関連するチートシートはココにまとめて置いてある。

### ∗ R Markdown ファイルを PDF ファイルに出力する

R Markdown を PDF ファイルに出力するときは、`rmarkdown::render()` を使う。

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。

```
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 18 行までがヘッダであり、そのうち、“pdf_document:” のブロックと、**documentclass:** 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: yes” 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: no” 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 等の PDF リーダで開いて確認してみよう。

Knit ボタンを使って DF ファイルを作ると、図の中の日本語が文字化けする（トーフになる）かもしれない。その場合、`knitr::render()` コマンドで PDF を作ると文字化けしない（かもしれない）。コマンドを使っ PDF ファイルを作るときは、`rmarkdown::render()` を使う。

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

これでも文字化けが解消しない場合は、以下の設定が必要である。

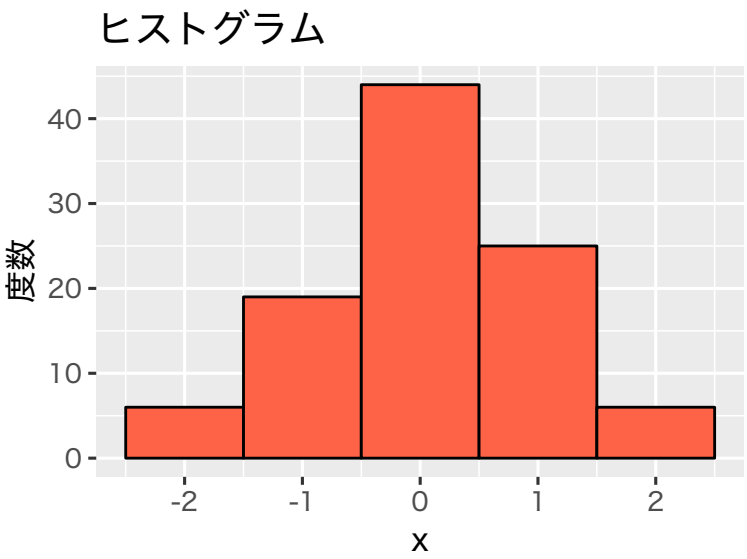
1. 日本語を使う図を表示するコードチャンクで、chunk オプション **dev = “cairo_pdf”** を指定する（Mac の場合は **dev = “quartz_pdf”** も使える）。

- すべてのチャンクに適用するなら、global_option で指定してもよい
- PDF 以外に出力する場合は、オプションを外したほうがよい（そのままにすると、HTML で図だけ PDF になってしまう [Couldn’t load plugin. と表示される。] ）

2. フォントを指定する

- Mac の場合：`par(family = "HiraginoSans-W3")`
- Windows の場合：`par(family = "Japan1GothicBBB")`

```
print(p)
```




詳しくは、ココやココを参照。

### R Markdown ファイルを HTML ファイルに出力する

R Markdown を HTML ファイルに出力するときは、`rmarkdown::render()` を使う。

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 18 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: yes” 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: no” 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）を押しても HTML ファイルを作れる。

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。

```
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しなくてよい。今はボタンで変換できればよい）。

## R Markdown の例：2 変数の記述統計

前回の授業で使った架空のデータセット fake-data-01.csv を使い、2 つの量的な変数の関係を調べてみよう。

前回同様、このデータセットを `myd` という名前で利用する。

```
myd <- read_csv("data/fake-data-01.csv")
```

```
## Parsed with column specification:
## cols(
##   id = col_integer(),
##   gender = col_character(),
##   age = col_integer(),
##   height = col_double(),
##   weight = col_double(),
##   income = col_integer()
## )
```

データを読み込んだら、中身を確認する。

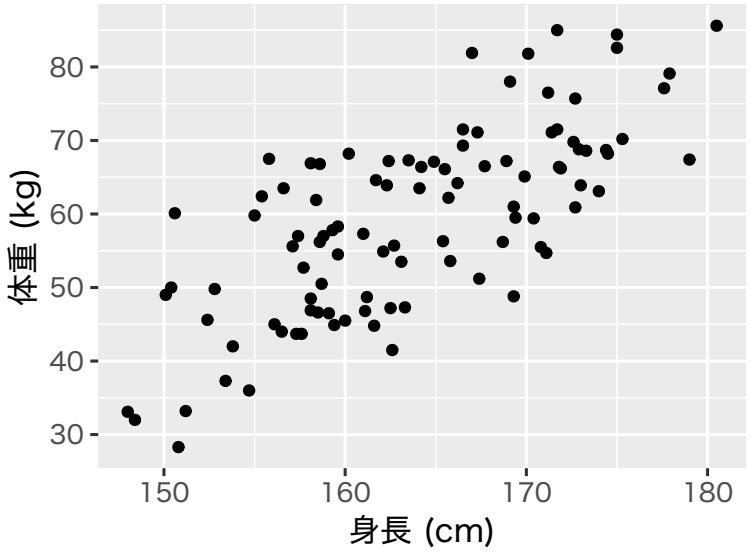
```
glimpse(myd)
```

```
## Observations: 100
## Variables: 6
## $ id     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ gender <chr> "male", "male", "male", "male", "male", "male", "male",...
## $ age    <int> 52, 33, 22, 33, 26, 37, 50, 30, 62, 51, 55, 36, 66, 42,...
## $ height <dbl> 174.0, 175.3, 175.0, 170.1, 167.4, 159.3, 173.3, 162.5,...
## $ weight <dbl> 63.1, 70.2, 82.6, 81.8, 51.2, 57.8, 68.6, 47.2, 68.2, 5...
## $ income <int> 3475810, 457018, 1627793, 6070642, 1083052, 2984929, 14...
```

## 2 つの量的変数の関係を図示する

2 つの量的変数の関係は、散布図 (scatter plot) で確認する。ここでは、身長 (height) と体重 (weight) の関係を図示してみよう。ggplot2 では、`geom_point()` で散布図ができる。

```
scat <- ggplot(myd, aes(x = height, y = weight)) +
  geom_point() +
  labs(x = "身長 (cm)", y = "体重 (kg)")
print(scat)
```




このデータセットに含まれる身長と体重の間には、どのような関係があるだろうか？

## 2 つの量的変数の関係を統計量で示す

2 つの量的変数の関係を表すのにもっともよく使われるのは、相関係数 (correlation coefficient) である。この統計量は、$r$ で表されることが多い。$-1 \leq r \leq 1$ となる。$a$ と $b$ という 2 つの変数があったとき、$a$ が大きくなるほど $b$ も大きくなるという関係があるとき、「$a$ と $b$ には正の相関 (positive correlation) がある」と言い、このとき $r > 0$ である。また、$a$ が大きくなるほど $b$ が小さくなるという関係があるとき、「$a$ と $b$ には負の相関 (negative correlation) がある」と言い、このとき $r < 0$ である。$r =$ のとき、「$a$ と $b$ は相関関係がない」と言う。

正の相関があるとき、$r$ が 1 に近いほど、その関係は強い。また、負の相関があるとき、$r$ が $-1$ に近いほど、その関係は強い。つまり、相関関係は、相関係数の絶対値が 1 に近いほど強い。

R で相関係数を求めるときは、$cor()$ を使う。身長と体重の相関係数は、

```
cor(myd$height, myd$weight)
```

```
## [1] 0.7294207
```

である。この 2 変数にはどんな関係があるだろうか？

## 散布図と相関係数

2 つの量的な変数の関係を調べるときは、散布図と相関係数の**両者**を使ったほうがよい。

散布図だけを使うと、本当は存在しない関係を、誤って見つけてしまうことがある。例えば、本当は相関がない 2 つの変数の散布図を描いたとき、描かれた点がなんとなく右肩上がりの直線の周りに集まっているように見えてしまうことがある。これは、人間がパタンを見つける能力に優れている（優れ過ぎている？）からだと考えられる。偶然できた壁のシミが人間の顔に見えてしまうことがあるというのも似たような現象である。

散布図だけに頼ると、存在しないパタンが見えてしまうことがあるので、散布図で発見したパタンが本当にあるかどうか、相関係数を求めて確かめるべきである。

反対に、相関係数だけに頼るのも危険である。相関係数は、2 変数のあらゆる関係を捉えられるわけではない。相関係数が示すのは、2 つの相関係数の**直線的な関係だけ**である。

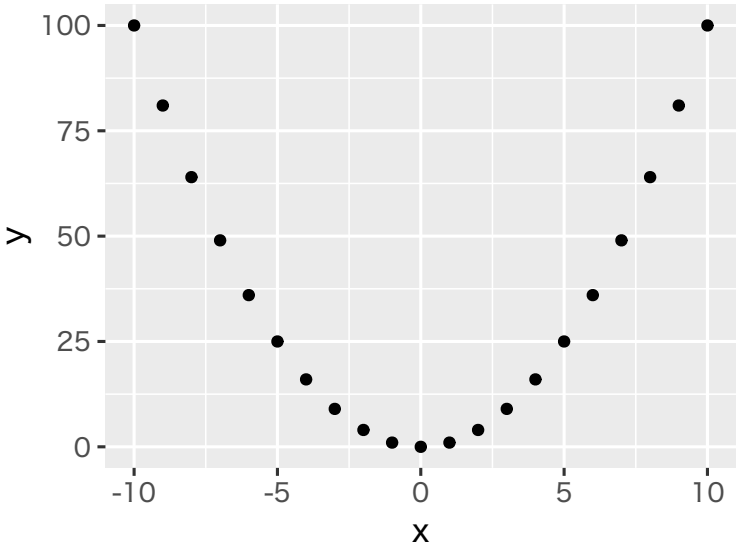
例として、$x$ と $y$ という 2 つの変数を以下のとおり作り、相関係数を計算してみよう。

```
x <- -10:10
y <- x^2
cor(x, y)
```

```
## [1] 0
```

2 変数と $x$ と $y$ の相関係数は 0 である。相関係数だけに頼ると、2 つの変数の間には関係がないと言う結論が出せそうである。しかし、相関係数が低くても、必ず散布図を描いたほうがよい。散布図を作ってみよう。

```
newd <- tibble(x = x, y = y)
scat2 <- ggplot(newd, aes(x = x, y = y)) +
  geom_point()
print(scat2)
```




この図を見て、$x$ と $y$ は無関係と言えるだろうか？

散布図から明らか（$y$ をどのように作ったかを見ればもっと明らかだが）なように、$x$ と $y$ には強い関係がある（$y$ は $x$ の関数である）。しかし、その関係は**曲線的**なので、**直線的な関係しか捉えられない相関係数**は、強い関係を見落としてしまうのである。

このように、2 つの量的変数の関係を調べるときは、**散布図と相関係数の両方を確認する**習慣を身につけたほうがよい。

授業の内容に戻る