



ミクロ政治データ分析実習

第4回 R Markdownと再生可能な研究

そん じえひよん

宋 財 泱

関西大学総合情報学部

2021/5/6 (updated: 2021-05-03)

始める前に

第3回の講義動画の内容について

- NIIオンライン分析システムを使用している場合、macOS、Windows関係なく、「NIIオンライン分析システム」の箇所を参照する。
 - こちらはクラウド版のR/RStudio
 - ここで作業したコード、図表、データなどはすべてサーバー上に保存され、自分のPCには保存されない。
 - 課題提出などのために自分のPCに保存する必要がある場合は、サーバーからファイルをダウンロードする（本日講義の後半に解説）。
- 自分のPCに直接R/RStudioをインストールした場合、自分のOSの箇所を参照する。

その他

- NIIオンライン分析システムを使用している場合、「初期設定」は1回のみ
 - 初期設定を行う度に新しいサーバーを立ち上げることになるので注意すること

R Markdownとは

R Markdownとは

- R + Markdown
- Rのコードと結果、そして文章が一つの文書としてまとめたもの
 - 文書の形式はHTML、Microsoft Word（.docx）、PDFで出力される
 - スライド、ホームページなども作成可能
 - 本講義のスライドもR Markdownで作成（{xaringan}）
 - 本講義の教科書もR Markdownで作成（{bookdown}）
 - 宋のホームページもR Markdownで作成（{blogdown}）

Markdownとは

プレーンテキストとリッチテキスト

- Plain text: 書式情報などが含まれていない純粋なテキストのみで構成されている文書
 - macOSの「テキストエディット」、Windowsの「メモ帳」など
- Rich text: 書式情報だけでなく、図表なども含まれる文書
 - Microsoft Word、Apple Pagesなど
- 書式情報の例: 文書の余白、行間の幅、文字サイズ、太さ、色
- リッチテキストには文字情報以外にも様々な書式情報が含まれているため、サイズが大きく、パソコンへの負担も大きい

プレーンからリッチへ

Markup言語とは

- プレーンテキストをリッチテキストへ変換するための言語
- HTMLが代表的
 - 例: と で囲まれた文字を太字として出力

Input

この文字は太字にしましょう。

この文字はリンクにしましょう。

Output

この文字は**太字**にしましょう。

この文字は[リンク](https://www.jaysong.net)にしましょう。

Markdownとは

- **Markdown**: **Markup**言語の軽量版
- 文章作成に必要なタグ (tag) をより簡単に記述できる。
- ブラインドタッチに慣れればWordよりも素早く文書を作成するようになる。

Markupの場合 (HTML)

この文字は太字にしましょう。

この文字はリンクにしましょう。

Markdownの場合

この文字は**太字**にしましょう。

この文字は[リンク] (<https://www.jaysong.net>)にしましょう。

Markup vs. Markdown

表の作成

Markup

```
<table>
  <thead style = "font-weight:bold"
    <td>ID</td>
    <td>Name</td>
  </thead>
  <tr>
    <td>1</td>
    <td>Song</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Kojima</td>
  </tr>
</table>
```

Markdown

ID	Name
1	Song
2	Kojima

Output

ID	Name
1	Song
2	Kojima

Markup vs. Markdown

箇条書き

Markup

```
<ul>
  <li>箇条書き1</li>
  <ul>
    <li>箇条書き1A</li>
    <li>箇条書き1B</li>
  </ul>
  <li>箇条書き2</li>
  <li>箇条書き3</li>
</ul>
```

Markdown

```
* 箇条書き1
  * 箇条書き1A
  * 箇条書き1B
* 箇条書き2
* 箇条書き3
```

Output

- 箇条書き1
 - 箇条書き1A
 - 箇条書き1B
- 箇条書き2
- 箇条書き3

とりあえずKnit

• Rmd ファイルの生成

- File > New File > RMarkdown ...
- 何も触らず、OKをクリック
- クラウド版RStudioの場合
 - Sourceペイン上段の から Preview in Viewer Pane を選択
 - チェックしない場合、結果ファイルが新しいウィンドウとして出力されるため、面倒
- ファイルを保存 (File > Save)
 - ⌘+S (macOS) 、または Ctrl + S (Windows)
- Sourceペイン上段の Knit をクリック
 - Knit: 「ニット」と読む

ABC Knit

```
1 title: "Untitled"
2 author: "Jaehyun Song"
3 date: "2/21/2021"
4 output: html_document
5
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for
15 authoring HTML, PDF, and MS Word documents. For more details on using R
16 Markdown see <http://rmarkdown.rstudio.com>.
17
18 ```{r cars}
19 summary(cars)
20 ```
21
22 ## Including Plots
23
24 You can also embed plots, for example:
25
26 ```{r pressure, echo=FALSE}
27 plot(pressure)
28 ```
29
30 Note that the `echo = FALSE` parameter was added to the code chunk to
31 prevent printing of the R code that generated the plot.
```

YAMLヘッダー

チャンク

見出し

本文

チャンク

見出し

本文

チャンク

YAMLヘッダー

Untitled

Jaehyun Song

2/21/2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

	speed	dist
## Min.	4.0	Min. : 2.00
## 1st Qu.	12.0	1st Qu.: 26.00
## Median	15.0	Median : 36.00
## Mean	15.4	Mean : 42.98
## 3rd Qu.	19.0	3rd Qu.: 56.00
## Max.	25.0	Max. : 120.00

Including Plots

You can also embed plots, for example:

pressure

YAMLヘッダー

.Rmd ファイル最上段に位置する文書の情報に関するデータ

```
---
```

```
title: "Untitled"
author: "Kandai Taro"
date: "2/21/2021"
output: html_document
```

```
--
```

- `title`: 文書のタイトル
- `author`: 文書の作成者
- `date`: 日付
 - 最後にKnitした日にする場合、

```
date: `r Sys.Date()`
```

- `output`: 出力形式
- その他オプション多数
 - YAMLヘッダーの詳細は[R Markdown Cheetsheet](#)などを参照

Markdownの文法

文章の改行

- 2回改行

Input

これは文章1です。

これは文章2です。

Output

これは文章1です。 これは文章2です。

Input

これは文章1です。

これは文章2です。

Output

これは文章1です。

これは文章2です。

強調

- 英語の強調はイタリック、日本語の強調は太字を使用

Input

文章の一部を**太字**にしてみましょう。

Canada, France, Germany, Italy, *Japan*, UK, and USA.

~~取り消し線~~はあまり使わないかも。

アンダーラインはHTMLタグを使います。

Output

文章の一部を**太字**にしてみましょう。

Canada, France, Germany, Italy, *Japan*, UK, and USA.

~~取り消し線~~はあまり使わないかも。

アンダーラインはHTMLタグを使います。

見出し

- # + 半角スペース + 見出し (#が多いほど小さくなる)

Input

```
# 見出し1 (ヘッディング1)
## 見出し2 (ヘッディング2)
### 見出し3 (ヘッディング3)
#### 見出し4 (ヘッディング4)
```

Output

見出し1 (ヘッディング1)

見出し2 (ヘッディング2)

見出し3 (ヘッディング3)

見出し4 (ヘッディング4)

水平線

- 任意の行に --- を挿入
- 章と章の区切り、節と節の区切りなどに使用

Input

文章1

文章2

Output

文章1

文章2

箇条書き（順序なし）

- * + 半角スペース + 文字列
- 階層構造にする場合、* の前に半角スペースを3個以上入れる
- * の代わりに - も可

Input

```
* リスト1
* リスト2
  * リスト2-1
    * リスト2-1-1
    * リスト2-1-2
  * リスト2-1
* リスト3
```

Output

- リスト1
- リスト2
 - リスト2-1
 - リスト2-1-1
 - リスト2-1-2
 - リスト2-1
- リスト3

箇条書き（順序付き）

- 1. + 半角スペース + 文字列
- 1. の箇所は任意の数字でもOK。1.、2.、...も可
- 階層構造にする場合、*の前に半角スペースを2個以上入れる

Input

```
1. リスト1
2. リスト2
  1. リスト2-1
    1. リスト2-1-1
    2. リスト2-1-2
  2. リスト2-1
3. リスト3
```

Output

```
1. リスト1
2. リスト2
  1. リスト2-1
    1. リスト2-1-1
    2. リスト2-1-2
  2. リスト2-1
3. リスト3
```

数式

- 文中数式は \$数式\$、独立した行の数式は \$\$ で囲む
- $LATEX$ の数式表記法を理解する必要がある

Input

文中の数式の例: ` \(\mathbf{e} = mc^2\)`

独立した数式の例は、

```
$$  
y_i \sim \text{Normal}(\mathbf{X} \boldsymbol{\beta}, \sigma).  
$$
```

Output

文中の数式の例: $e = mc^2$

独立した数式の例は、

$$y_i \sim \text{Normal}(\mathbf{X}\boldsymbol{\beta}, \sigma).$$

表

- 行は改行で、列は | で区切る
- 1行目は表のヘッダーであり、|---| で区切る (- は1個以上でOK)
 - |---| は左揃え、|---:| は右揃え、|:---:| は中央揃え

Input

ID	Name	Birth	Pref
1	Ogura	1852	Tokyo
2	Tsurumi	1858	Tochigi
3	Yoshida	1858	Fukuoka
4	Hotta	1859	Chiba
5	Inoue	1847	Nagano
6	Tezuka	1862	Tokyo
7	Shikata	1857	Saitama
8	Nomura	1855	Gunma

Output

ID	Name	Birth	Pref
1	Ogura	1852	Tokyo
2	Tsurumi	1858	Tochigi
3	Yoshida	1858	Fukuoka
4	Hotta	1859	Chiba
5	Inoue	1847	Nagano
6	Tezuka	1862	Tokyo
7	Shikata	1857	Saitama
8	Nomura	1855	Gunma

コメント

出力に影響されないコメントを入れる場合

- <!-- と --> 内に囲む
- <!-- と --> の間の文章は出力されない
- 自分用のメモ、とりあえず出力させない内容などに使用
- チャンク内コメントは # (チャンクは後に解説)

Input:

```
これは文章1です。
```

```
<!--
```

```
これはコメントです。
```

```
-->
```

```
これは文章2です。
```

Output:

```
これは文章1です。
```

```
これは文章2です。
```

その他の文法について

教科書第25章を参照

- 図の挿入
- 脚注
- 引用など
- HTMLタグを直接入れることも可能
 - 出力形式がHTMLならJavaScriptも可
- [R Markdown Cheatsheet](#)も参照

RとMarkdownの結合

チャンク (Chunk)

Rのコードを挿入する領域

- ``{r} と ``` で囲む
 - チャンクの外にコードを書いても文章として認識する
- コードの出力 > 結果の出力の順番

Input

ここはチャンクの外

ここに書いたものは文章として扱われる。

```
``{r}  
x <- 5 # 円の半径  
print(3.14 * x^2)  
```
```

ここに書いたのも文章

# 結果

- コードと結果が同時に出力される

## Output

ここはチャンクの外

ここに書いたものは文章として扱われる。

```
x <- 5 # 円の半径
print(3.14 * x^2)
```

```
[1] 78.5
```

ここに書いたのも文章

# チャンクのオプション

{r} の箇所にオプションが指定可能 ( , の後)

- チャンクにDay4-1という名前を付ける
  - チャンク名の場合 , は不要。オプションはチャンク名の後に付ける
  - {r Day4-1}
- コードのみ出し、結果は出力しない
  - {r, eval = FALSE}
- 結果のみ出し、コードは出力しない
  - {r, echo = FALSE}
- 出力される図の幅を8インチ、高さを10インチに
  - {r, fig.width = 5, fig.height = 10}
- 出力される図を中心揃えに
  - {r, fig.align = center}
- 警告メッセージを隠す
  - {r, warning = FALSE}
- 他にも数十のオプションの指定が可能
  - 詳しくは[R Markdown Cheatsheet](#)を参照

# インラインコード

文章内にRコードの結果を埋め込む場合

- ``r と `` の間にコードを挿入

**Input:**

```
```{r}
x <- 5 # 円の半径
```

```

円の半径は`r x`、面積は`r 3.14 \* x^2`です。

**Output:**

```
x <- 5 # 円の半径

```

円の半径は5、面積は78.5です。

# R Markdownと再生可能な研究

---

# 再現可能性と再生可能性

国里(2020)<sup>1</sup>による定義

- 学会、分野によっては正反対の概念になる場合もある (Barba, 2018<sup>2</sup>)

再生可能性 (Reproducibility)

- 同じデータ + 同じ方法 = 同じ結果
- R Markdownの主な目的は**再生可能性**

再現可能性 (Replicability)

- 新しいデータ + 同じ方法 = 同じ結果
- 新しいデータ + 違う方法 = 同じ結果

[1] 国里愛彦. 2020. 「再現可能な心理学研究入門」 『専修大学人間科学論集:心理学篇』 10. 21-33.

[2] Lorena A. Barba. 2018. "Terminologies for Reproducible Research." *ArXiv*. 1802.03311

# R Markdownと再生可能性

1. 具体的にどのような手順で分析が行われたかが分かる
  - コードを記録&公開
2. コードに対する説明が付いている
  - 自分が書いたコードであっても、昔書いたコードは読みにくい
  - 過去、自分が行った分析の再生する出来ない場合も
    - 今、教壇に立っている人の話ではない
3. コードと結果が同時に照会できる
  - わざわざRを立ち上げなくても、その場で分析の手順が分かる

# 再生可能な研究を目指して

- 可能な限り、詳細な説明を付ける
  - 文章領域では「以下のコードがどのような目的で書かれたコードであるか」などを明示する。
  - チャンク内にもこまめにコメント（#）を付ける
    - 理想としては一行ごとにコメントを付ける

## 悪い例

```
x <- rep(NA, 5)
```

## 良い例

```
x <- rep(NA, 5) # 欠損値で埋められた長さ5のベクトル
```

# 再生可能な研究を目指して

- コードを分かりやすく書く
  - 長いオブジェクト名を避ける
  - 分かりやすいオブジェクト名
  - 適宜スペースと字下げを入れるなど
  - 読みやすいコードの書き方: [教科書 第11.2章](#)

---

例1    例2    例3    例4a    例4b

## 悪い例

```
5人の数学成績
variable1 <- c(30, 91, 43, 77, 100)
```

## 良い例

```
mathematicsscore <- c(30, 91, 43, 77, 100)
```

# 再生可能な研究を目指して

- データについても詳細に書く
  - どのように収集されたデータであるのか
  - どのような変数があるのか

# まとめ

---

# 今回の内容

よく分からぬ箇所は教科書を読み返す or 宋&TAに質問 (できれば、LMSで)

- R Markdown: 第25章

# 課題

今回以降の練習はR Markdownを利用する。

1. 今回講義用のプロジェクトを作成する。
2. LMSから課題用資料の「問題」と「サンプル」をダウンロードし、プロジェクトのフォルダーに保存する。
  - NIIオンライン分析システムの場合、サーバーにアップロード
  - **ファイル名は変更しないこと**
3. プロジェクトからRStudioを起動し、`.Rmd` ファイルを開く
4. Knitし、結果を確認する。
  - Source Pane上段のKnitをクリックするか、「⌘ + Shift + K」(macOS)、「Ctrl + Shift + K」(Windows)を推す。
5. サンプルファイル (`MicroAnalysis_Day04_Answer.html`) を開く。
6. サンプルのような体裁になるように問題ファイルを修正する。
  - 隨時Knitしながら結果を確認する。
7. 完成後、`.Rmd` ファイルと生成された`.html` ファイルを関大LMSに提出する。
8. **期限は2021年5月6日（講義日）の23時59分とする。**
9. 答案は次回の講義までに公開する。

# 課題提出の方法

NIIオンライン分析システムの場合、`.Rmd` ファイルと `.html` ファイルを自分のPCにダウンロードする必要がある。

- NIIオンライン分析システムはクラウド版のR/RStudioであるため、すべての作業物はサーバー上のみに記録される。

## 方法

1. jupyter hub画面でダウンロードするファイルの左のチェックボックスにチェック（☑）を入れる。
2. 上段の「Download」をクリックする。
3. ダウンロードしたファイルを提出する。