

# 応用プログラミング3 第3回 R Markdown によるレポート作成

専修大学ネットワーク情報学部  
田中健太

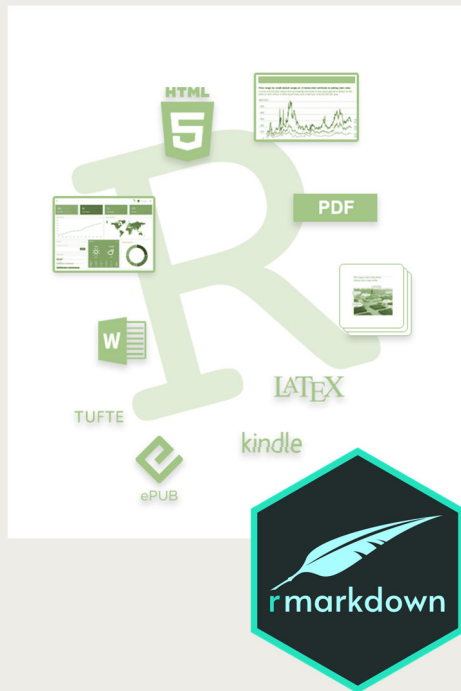
# 課題の振り返り

# 1. R Markdownの概要

2

今回の講義では、Rやその他の言語で作成したプログラムと、分析結果の解釈や考察などの文章を混在させ、技術文書（ドキュメント）を作成するためのフレームワークである、R Markdownについて紹介します。

## 1.1 R Markdownとは



- R Markdownは、RプログラムとMarkdownによるテキストを混在させたドキュメントを作成できる仕組み
- HTML, Word, PowerPoint, PDFなどさまざまなフォーマットで出力できる
- Rだけでなく、他のさまざまな言語に対応しており、汎用の科学技術文書作成プラットフォームとして利用が広がっている

3

R Markdownは、rmarkdownおよび関連パッケージが提供する、ドキュメント作成のフレームワークです。ひとつのファイルに、分析の目的や現状といった文書と、実際の分析プログラムを混在させ、さまざまなフォーマットで出力できます。

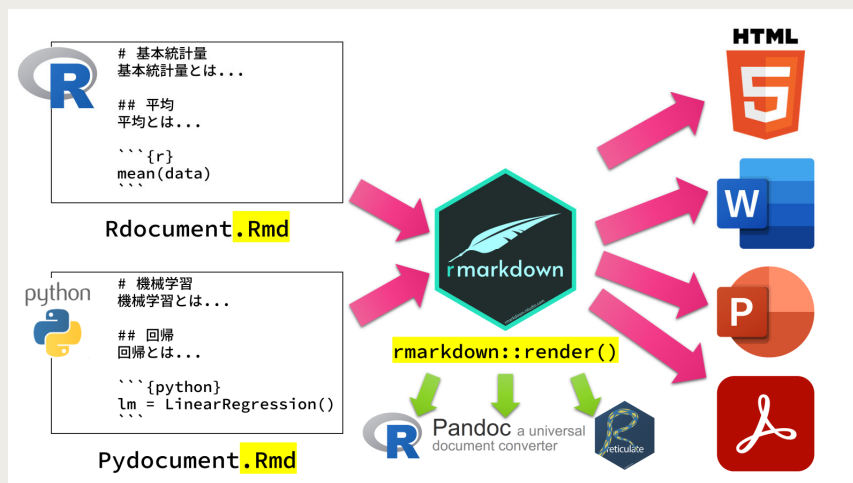
一般に、データ分析とその背景や結果の解釈を述べた文書は、セットで作成します。研究論文でも、ビジネスにおける報告書でも同じです。しかし、文書をWordやPowerPointで作成し、データ分析を独立したRやPythonのファイルで行うと、データやパラメータを変えて再度分析した結果を、文書に反映し忘れた（グラフが古い）、といったことが起こりやすくなります。そのようなことを避けるためにも、文書とプログラムは混在することが望ましく、古くから「文芸的プログラミング」という考えがありました。これを提唱したD. E. Knuthは、TeX (LaTeX) という組版処理システムを開発し、現在でも論文執筆におけるスタンダードなツールとして使われています。

Rでも、以前から文芸的プログラミングの考えに沿ったSweaveという仕組みがあります。しかし、記法が独特で、LaTeXに習熟していないと活用できないため、ユーザーが限られていました。

一方で、IT業界ではMarkdownという文書フォーマットが登場し、広く使われるようになっていました。そのような流れの中で、2012年に、knitrというパッケージが公開されました。これは、Sweave以外にも、HTMLやReST (reStructuredText)、そしてMarkdownで記述した文書に、Rプログラムを組み込むことができ、さまざまなフォーマットに変換して出力できる、画期的なものでした。knitrパッケージが対応するフォーマットの中で、特にMarkdownについてのニーズが多く、knitrから切り出して強化された、rmarkdownパッケージが提供されています。現在では、R Markdownはデータ分析の結果をレポートするための標準的なフォーマットになっています。

## 1.2 R Markdownの処理フロー

- R Markdownドキュメントの拡張子は **.Rmd**
- Markdownと**チャンク**を組み合わせて記述する
- .Rmd ファイルから各フォーマットへの変換は **rmarkdown**パッケージの **render()** 関数で行う



4

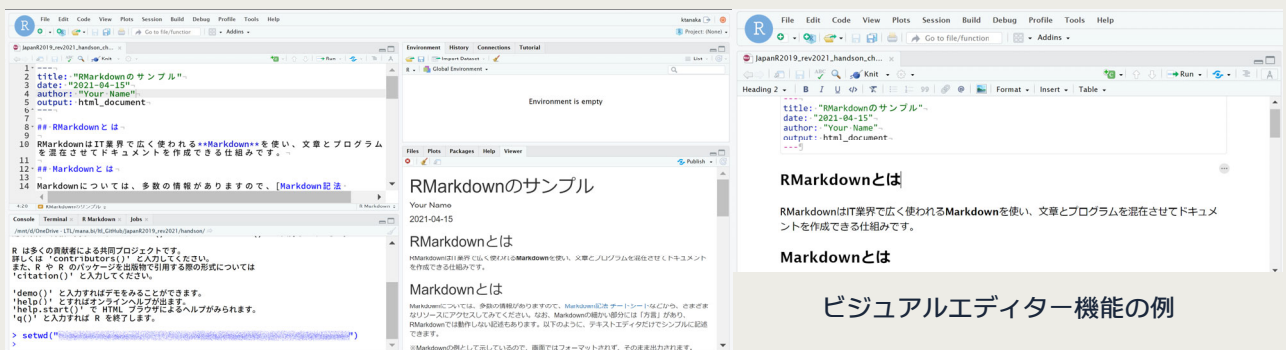
R Markdownによるドキュメント作成の、全体的な流れを示します。まず、RStudioやその他のエディターで、R Markdownファイルを作成します。拡張子は .Rmd です。ファイルの内容については、この後述べますが、Markdownによるドキュメントと、チャンクと呼ばれる要素の中に、RやPythonのプログラムを記述します。

そして、R Markdownファイルをさまざまなフォーマットに変換するのは、rmarkdownパッケージの render() 関数です。関数を実行するか、RStudioのメニューボタンを押すことで、R Markdownファイルに書かれた指定に合わせて、変換が行われます。詳しく処理の流れを記すと、render() 関数を実行すると、Rがチャンクに書かれたプログラムを実行します。そして、その結果を含めたMarkdownファイルが生成されます。

さらに、Markdownファイルを、Pandocという汎用フォーマット変換ソフトウェア（後述）が、HTMLやPowerPoint形式に変換します。

## 1.3 R Markdownの作成環境

- `install.packages("rmarkdown")` でインストールする
- Jupyter (JSON形式) と異なり、プレーンテキストなので、どんなエディターでも作成できる
- RStudioやVisual Studio Codeには、R Markdownの作成支援機能がある
- RStudioでは、ビジュアルエディター機能も使用できる



5

R Markdownファイルを作成するには、`rmarkdown`パッケージが必要です。  
`install.packages()` 関数か、RStudioのメニューからインストールします。R Markdown形式のファイルは、プレーンテキストなので、RStudioでもその他のエディターでも作成できます。この講義では、RStudioを使います。Pythonなどで広く使われる、Jupyter Notebook (Google Colab) の `.ipynb` ファイルは、中身はJSON形式なので、直接編集することは困難ですが、R Markdownはどのようなエディターでも編集できます。

また、RStudioをはじめ、多くのエディターに、R Markdownファイルの作成支援機能があり、コマンドの補完などができます。そして、比較的新しい (2020年以降の) RStudioでは、Visual Editorモードがあり、Wordやブログツールと同様の感覚で、見出しや画像の挿入、テキストの装飾ができます。

## 2. Markdownの基礎

6

R Markdownは、RプログラムとMarkdown文書の組み合わせなので、Rだけでなく、Markdownについても理解しておく必要があります。ここでは、Markdownによる文書作成の基本を紹介します。

## 2.1 Markdownとは

# 見出しレベル1

## 見出しレベル2

これは本文です。[リンク](http://example.com/)を参照してください。

![画像](./example.png)

- シンプルな記法で記述したプレインテキストをさまざまなフォーマットに変換できる

**マークアップ言語**の一種

- 2000年代はじめに開発がはじまり、現在では数多くのソフトウェア、Webサービスが対応しており、**技術文書のフォーマットとして標準になっている**

- 記法は、電子メールにおいてテキストを修飾する際の慣習に基づいている

7

Markdownは、シンプルな記法で文書の構造を記述する、テキストファイルのフォーマットです。当初、テキストファイルを、HTMLに変換するためのプログラムとして、Perl言語で開発されました。その後、Perl, HTMLに限定されず、さまざまな言語による実装と、多様なフォーマットに対応した変換ツール（コンバーター）が開発され、提供されています。プレインテキストで文書の構造を記述する仕組みは数多くありますが、結果として、Markdownが世界的に広く受け入れられ、現在では、IT業界における技術文書の標準フォーマットと言えるほど、普及しています。Markdownは、GitHubやQiitaなどのエンジニアが利用するWebサービスでも、標準の文書フォーマットになっています。

なお、Markdownには厳密な仕様や、統一的な管理組織が存在しないため、実装によって細かな違い、「方言」があります。そのため、あるサービスで利用できる記法が、別のサービスでは使用できない、といったことがあります。インターネットの「Markdownの使い方」といったような記事で紹介されている記法が、R Markdownでは動作しないこともありますし、逆にR Markdown独自の拡張も存在します。今回は、R Markdownで利用できる記法に限って紹介します。



## 2.2 見出しと段落

- **見出しは # で指定する**
  - # の数を増やすごとにレベルが下がる
  - レベル1-6まで対応している（HTMLと同じ）
  - 出力形式によって、特に#と##の扱いが異なる
  - #と見出し項目の間には半角スペースを入れる
- 空行を挟むことで改段落できる
  - 段落の先頭の字下げは、全角スペースなどではなく、テンプレートの設定で行う

8

文書の構造を表現する、代表的なものに見出しと段落があります。特に、文学作品ではない、論文やレポート、ビジネスにおける報告書などの技術文書では、見出しによる構造化が必要です。

Markdownでは、# を使い、見出しを表現します。HTMLの `<h1>` から `<h6>` に対応し、# の数を増やすほど、見出しのレベルが下がります。また、# と見出し項目の間には、半角スペースを入れる必要があります。なお、R Markdownでは、最終的な出力フォーマットによって、# (`<h1>`) と ## (`<h2>`) の扱いが変わります。HTMLでは、一般に `<h1>` がタイトルを表しますので、文書の内部構造としての見出しは、## から始めます。一方、WordやPowerPoint、PDF (LaTeX) では、タイトルは別途出力され、# が章レベルの見出しとして扱われます。

もう一つ、文書の構造として大事なものに、段落があります。Markdownでは、空行を挟むことで改段落します。空行のない改行は、無視され、続けて出力されます。なお、日本語（の、特に印刷媒体）では、段落の先頭を1文字下げる字下げ・インデントが一般的ですが、標準では、R Markdownの出力で字下げはされません。全角スペースを入れるのではなく、HTMLの場合はCSSで、Wordではテンプレートの設定で、字下げを指定します。

## 2.3 箇条書き

### 1. 順序付き箇条書き

#### 1. 番号は自動的に増加する

- 1. サブ項目はタブでインデント

#### \* 順序なし箇条書き

- \* サブ項目はタブで<br>インデント

#### - 順序なし箇条書き

- 箇条書きは順序付き（HTMLの`ol`）と順序なし（`ul`）を作成できる

- 順序付き箇条書きは、自動的に番号を増加（インクリメント）してくれる

- レベルを下げたサブ項目は、タブ（または半角スペース4個）でインデントする

- HTMLタグは機能する出力としない出力がある

箇条書きは、いくつかの事柄を、順序立ててまたは並列に表現するときに使います。Markdownでは、HTMLの `<ol>`, `<ul>` に対応する箇条書きを記述できます。

1, 2, 3… と順序のある箇条書きは、行頭に 1. と記述すると、自動的に番号が増加（インクリメント）します。コンマの後ろには、半角スペースを入れます。空行を挟むと、番号がリセットされます。

順序のない箇条書きは、行頭に \* または - を記述します。また、順序あり、なしいずれについても、1段階レベルを下げたサブ項目を作成できます。この場合は、行頭にタブまたは半角スペース4個を入れます。

## 2.4 リンクと画像の挿入

日々のデータは[ダッシュボード](https://example.com/dashboard)で確認できます。

以下の図は、昨日の分析結果です。

![結果01](./example01.png)

```
<div align="center">  
  
</div>
```

- リンクは [文字列](URL) という形式で記述する
- HTMLの <a> タグを使ってもよいが、出力フォーマットによっては正常に動作しない可能性がある
- 画像は ![代替文字列](URL) という形式で記述する
- 標準ではサイズが画面の横幅一杯になるので、細かい調整がしたい場合は、HTMLの <img> タグなどを使う

10

次に、文書中でのリンクや画像の挿入について紹介します。URLをそのまま記述すると、自動的にリンクが貼られます。また、文字列にリンクを設定したい場合は、[リンク文字列](URL) とします。HTML出力で、リンクを別タブで開きたい場合は、[リンク文字列](URL){target="\_blank"} とします。また、target="\_blank" という指定は、近年セキュリティ上問題があるということで、それを考慮して [リンク文字列](URL){target="\_blank" rel="noopener noreferrer"} としてもよいでしょう。

- 参考：noopener と noreferrer の整理、結局どっちを使えば良いのか  
<https://blog.ojisan.io/noreferrer-noopener>

画像も、リンクと同じような書式ですが、冒頭に！をつけて、![代替文字列](URL) とします。URLは、インターネット上のものでも、ローカルのファイルパスでも構いません。ただ、インターネット上の画像は削除されることもありますし、変換処理のたびにWebにアクセスしてダウンロードすることになり、効率が悪いので、本当に必要なものはローカルにダウンロードして、それを参照するようにしましょう。なお、![ ]( ) の書式で挿入する画像は、サイズが画面(用紙)の横幅一杯になります(2段組みの場合は、1段分)。HTML出力の場合は、{width=50%} といったように幅を指定できます。WordやPowerPointでは指定できないため、出力されたファイルでサイズを変更します。

## 2.5 表組み

- Markdownにおける表は、縦棒 (|) と横線・ハイフン (-) で罫線を表現して作成する
- 1行目に列名を記述し、2行目に各列における文字の配置を : で指定する
- :--: は中揃え、:-- は左揃え、--: は右揃え
- セル結合など凝ったことはできないので、HTMLの <table> タグで記述する
- Rのデータフレームなどは、自動的に整形された表として出力される（後述）

11

Markdownでは、表も挿入できます。ただ、Excelのようにマウスで操作して作成することはできず、縦棒 (|) と横線・ハイフン (-) からなる罫線で、構造を表現します。

なお、実際にはある程度大きな表を手作業で作成することは少なく、Excelなどで作成したデータを、Excelのアドインや、エディターの変換機能でMarkdown形式にすることが多いでしょう。また、RStudioには、表の作成を支援するアドインがあり、インストールすることで、Excelからのコピーアンドペーストなどで、容易に表を挿入できます。

- 「CopyToMarkdownAddIn」 「Excel」のセルをMarkdown記法のテーブルとしてクリップボードへコピー - 窓の杜  
<https://forest.watch.impress.co.jp/library/software/copmarkdown/>
- inserttableアドイン: [https://github.com/lbusett/insert\\_table](https://github.com/lbusett/insert_table)

アドインはGitHub上で公開されているので、`remotes::install_github("ユーザー名/リポジトリ名")` 関数でインストールします。アドインは、RStudioの "Addins" メニューから起動します。なお、作成した表は、Rプログラムとして出力されるので、R Markdownのチャンクの中でアドインを実行する必要があります。

## 2.5 表組み

以下に分析結果の集計表を示します。

	ビール	ワイン	日本酒	ウイスキー	合計
---	:---	:---	:---	:---	:---
男性	40	15	25	20	100
女性	15	35	35	15	100

	ビール	ワイン	日本酒	ウイスキー	合計
男性	40	15	25	20	100
女性	15	35	35		

### 5.1 Markdownによる表組

	ビール	ワイン	日本酒	ウイスキー	合計
男性	40	15	25	20	100
女性	15	35	35	15	100

↑HTMLの出力例  
PowerPointの出力例→

# 3. R Markdownの基礎

13

ここからは、R Markdownに特有の記法、変換の方法について紹介します。

## 3.1 R Markdownファイルの全体像

- R Markdown (.Rmd) ファイルは、以下の要素から構成される
- ヘッダー (YAML)
- セットアップチャンク
- 本文 (Markdown)
- コードチャンク

```
---
title: "RMarkdownのサンプル"
date: "2021-07-10"
author: "Your Name"
output: html_document
---
```

ヘッダー

```
```{r setup, include = FALSE}
library(tidyverse)
options(dplyr.width = Inf, scipen = 1,
        digits = 4)
```

セットアップ  
チャンク

```
# R Markdownの基礎
```

```
## チャンクとは
```

```
RMarkdownでは、**チャンク**という要素にRプログラムを  
記述すると、プログラムを実行し、結果をドキュメントに含める  
ことができます。チャンクは、以下のように記述します。
```

Markdown  
本文

```
```{r eval=TRUE, message=FALSE,  
     warning=FALSE}  
a <- 1:10  
a  
```,
```

チャンク

14

繰り返しになりますが、R Markdownファイルの拡張子は、.Rmd です。ファイルには、大きく分けて4つの要素が含まれます。

1. ヘッダー (YAML)
2. セットアップチャンク
3. Markdownによる本文
4. プログラムを記述するチャンク

本文は文章なので自由に記述しますが、それ以外の要素は、ルールに沿って記述しないと、エラーになります。以降で、個別の要素について見ていきましょう。

## 3.2 ヘッダー (YAML)

```
---
title: "タイトル"
date: "日付"
author: "著者名"
output:
  html_document:
    toc: true
    number_sections: false
    pandoc_args: "--
highlight-style=tango"
---
```

- 文書全体の設定を **YAML形式** で記述する
- 基本的には 設定項目: 値 という形式で指定する
- 半角スペースによるインデントが厳密に評価される
- output 項目に出力フォーマットを (複数) 指定する
- 著者名など、複数項目を設定する場合は、- でリストとして記述する

15

R Markdownファイルの先頭には、ドキュメントのタイトルや著者、出力の設定などを記述したヘッダーが必要です。ヘッダーは、YAMLという形式で記述します。ハイフン3つ(---) で上下を囲む必要があります。設定項目は、項目名: 値という形式で記述します。また、インデントが厳密に評価されるため、タブの数が (本来のレベルより) 足りないと、R Markdownからの変換時にエラーになります。

YAMLで指定可能な、主な項目を列挙します。なお、output で指定する出力フォーマットについては、後述します。

- title: タイトル
- subtitle: サブタイトル
- author: 著者。複数人の場合は、author:(改行) - 1人目 (改行) - 2人目 と記述する
- date: 日付。直接文字列を指定したり、2022-09-22と現在の日時を指定できる

その他に、出力フォーマットに依存して指定可能な項目が多数あります。



## 3.3 チャンク

- プログラムと実行結果を出力するための特別な「囲み」がチャンク
- ```{エンジン名 オプション} ... ``` という形式で記述する
- R以外のエンジン (Python, RDBMS) は、対応するパッケージのインストールや設定が必要  
( <https://bookdown.org/yihui/rmarkdown/language-engines.html> )
- オプションは何も指定しなくてもよいが、`message=FALSE` と `warning=FALSE` は指定すると出力に余計なものが混じらず、見やすくなる
- コードを実行する場合は `eval=TRUE` が必須

16

R Markdownの最も重要で、便利な機能と言えるのが、ドキュメントの中でプログラムを実行できる、チャンク (chunk) の仕組みです。元々、Markdownでは、プログラムなどを表示するために、HTMLの `<pre>` に対応した、バッククォート3つで囲う記法があります。Markdownでは等幅フォントで表示するだけですが、R Markdownではこれを実行できます。

チャンクの基本的な書式は以下のようになります。

```
``{エンジン名 オプション1=値,オプション2=値,...}  
プログラム  
``
```

エンジンについては、R以外にも多様な実行環境がサポートされていますが、R以外については、別途実行環境を用意しておく必要があります。

- 参考: 2.7 Other language engines | R Markdown: The Definitive Guide  
<https://bookdown.org/yihui/rmarkdown/language-engines.html>

## 3.3 チャンク

```
```{r eval=TRUE, message=FALSE, warning=FALSE}
vec <- rnorm(10)
mean(vec)
```
```

↑ Rプログラムの例    ↓ Pythonプログラムの例

```
```{r eval=TRUE, message=FALSE, warning=FALSE,
engine="python"}
import numpy as np
vec = np.random.randn(10)
np.mean(vec)
```
```

17

オプションも、エンジンに依存する部分がありますが、共通してよく指定する項目に以下のようなものがあります。

- `eval=TRUE / FALSE`: チャンクの内容を実行するかどうか
- `message=TRUE / FALSE`: プログラムが出力するメッセージ (ユーザーが意図しないもの) を表示するかどうか
- `warning=TRUE / FALSE`: プログラムが出力する警告を表示するかどうか
- `echo=TRUE / FALSE`: 実行結果を出力するかどうか
- `cache=TRUE / FALSE`: 実行結果をキャッシュするかどうか

また、グラフィックスを出力するプログラムを含むチャンクでは、以下の項目でサイズを指定できます。

- `fig.width=サイズ`, `fig.height=サイズ`: グラフィックスの幅と高さを指定する。デフォルトは両方とも7 (インチ)
- `fig.cap=キャプション`: グラフィックスのタイトルを指定する
- `fig.align=配置`: グラフィックスの配置を指定する。`left`, `center`, `right` のいずれかを指定する

## 3.4 データフレームの出力について

- チャンク内でデータフレームを出力すると、標準ではRコンソールと同様の見た目になる
- それぞれの出力で見栄えの良い「表」として出力するには、パッケージの活用などが必要
- HTMLではヘッダーに `df_print: paged` などと指定する
- Word, PowerPoint, PDFでは出力ごとに最適な指定が異なる
- **flextableパッケージを使うのがいちばん無難か**

```
library(knitr)
kable(iris)
```

```
library(flextable)
flextable(iris)
```

18

読み込んだファイルの内容を確認したり、分析結果を示すために、データフレームを表示したいことがあります。R Markdownでは、Rスクリプト (.R) やコンソールでの作業と同じく、オブジェクト名を記述すると、オブジェクトの内容が出力されます。ただし、そのようにして得られる出力は、Rコンソールと同じ表示となり、論文やレポートなどのドキュメントとして、あまり見た目が良くありません。

HTML出力の場合は、ヘッダーに `df_print: paged` と記述すると、件数が多いデータフレームを、ボタン操作でページ切り替えして表示する出力が得られます。

Word, PowerPointなどについては、それぞれに適した表示でデータフレームを出力するためのパッケージ、関数が提供されています。knitrパッケージの `kable()` 関数では、シンプルな出力結果が得られます。kableExtraパッケージでは、`kable()` 関数の出力を拡張し、より柔軟な設定ができます。また、flextableパッケージは、HTML, Word, PowerPointいずれにおいてもほぼ同じ、整った出力が得られます。

## 3.5 数式の記述

- R MarkdownではLaTeXの記法で数式を記述できる

- 数式は  $$  で挟むとインライン（同一行内）に出力される

- 回帰モデル:  $y = a_1x_1 + a_2x_2 + \dots a_nx_n + b$

- $$  で挟むと別行立てで出力される

- 数式番号をつけて相互参照するなど凝ったことをしたい場合、LaTeXの知識が必須

...これについては式  
`\@ref(eq:model)`で示される。

```
\begin{equation}
y = a_{1}x_{1} + a_{2}x_{2}
+ \dots a_{n}x_{n} + b
(\@eq:model)
\end{equation}
```



...これについては式(1)で示される。

$$y = a_1x_1 + a_2x_2 + \dots a_nx_n + b \quad (1)$$

19

データ分析などのレポートには、数式がつきものです。(実際には、ビジネスにおいて報告書に数式が入っていると、「わからん!」とキレる偉い人もいますが) R Markdownでは、コンピューターで数式をマークアップする際の標準となっている、LaTeX形式で数式を記述できます。LaTeX記法についてここでは触れませんが、データサイエンティストや研究職を目指す方には必須のスキルと言えます。

なお、数式番号については、rmarkdownパッケージを拡張したbookdownパッケージを利用し、LaTeXのequation環境やalign環境とR Markdown特有のラベル (`\@eq:ラベル`) と相互参照 (`\@ref(eq:ラベル)`) 記法を組み合わせることで実現できます。ただし、これはbookdownパッケージの独自拡張なので、YAMLヘッダーに指定する出力形式も、bookdownパッケージが提供する形式 (例えば `bookdown::html_document2` や `bookdown::powerpoint_presentation2` など) にする必要があります。

- 参考1: 2.5 Markdown syntax | R Markdown: The Definitive Guide:  
<https://bookdown.org/yihui/rmarkdown/markdown-syntax.html#math-expressions>
- 参考2: 2.2 Markdown extensions by bookdown | bookdown: Authoring Books and Technical Documents with R Markdown  
<https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html>

## 3.6 コンパイル

- R Markdownファイルを所望のフォーマットで出力するには、RStudioの "knit" ボタンか、`rmarkdown::render()` 関数を使う
- RStudioでは "knit" ボタンの横の歯車ボタンで、出力をViewerペインに表示するか、別ウィンドウで表示するかなどを設定できる
- 複数の出力フォーマットを指定している場合は、先頭に書いたものが優先されるが、`render(..., output_format = "フォーマット名")` と選択できる

20

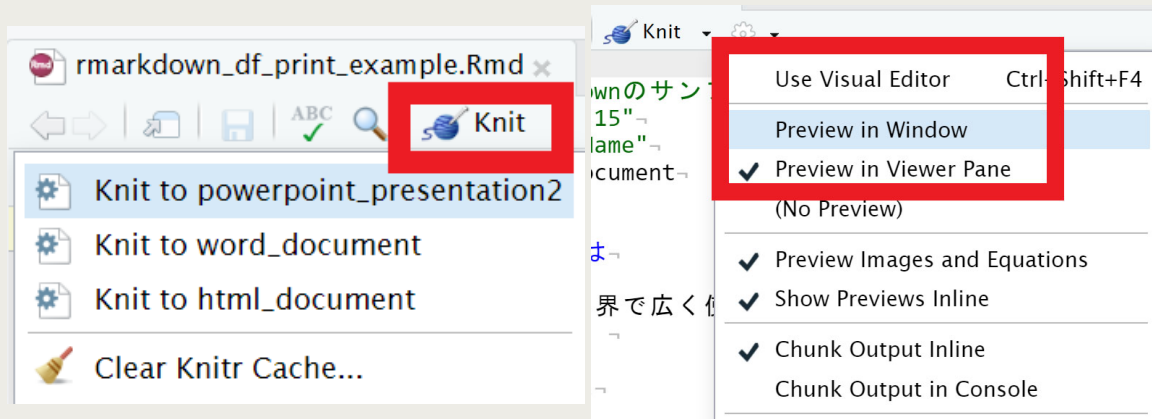
ここまで、R Markdownにおけるさまざまな記法について見てきました。まだ、出力フォーマットとオプションについては紹介していませんが、目的のフォーマットに変換、出力する手順を先に紹介します。

R Markdownファイルを任意のフォーマットに変換して出力するには、`rmarkdown`パッケージの `render()` 関数を使います。引数に `.Rmd` ファイルのファイル名を指定します。デフォルトでは、ヘッダーの `output_format` に最初に記述されたフォーマットに変換されます。他のフォーマットで出力したい場合は、`output_format = "フォーマット名"` と指定します。例えば、`output_format = "word_document"` などとします。また、複数のフォーマットを指定していて、すべて変換、出力したい場合は、`output_format = "all"` とします。

`render()` 関数による変換は、RStudioではメニューボタンとして用意されているので、"knit" ボタンを押せば実行できます。そして、HTMLに限りますが、出力をViewerペインに表示するか、別ウィンドウで表示するかを、横の歯車ボタンで設定できます。

なお、HTML以外の出力形式については、変換後のファイルに対応するアプリケーション (Acrobat Reader, PowerPointなど) で開いている時に、ドキュメントを修正するなどして、再度 `render()` 関数で変換しようとする、アプリケーションがファイルをロック (確保) しているため、エラーになります。変換時は、必ずファイルを閉じてから `render()` 関数を実行します。

## 3.6 コンパイル



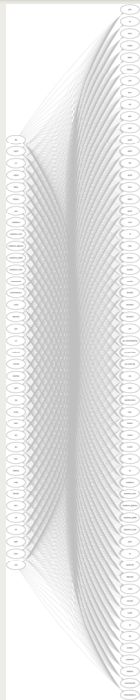
```
rmarkdown::render("ファイル名")
```

## 4. RMarkdownの出力

22

ここまで、R Markdownにおけるドキュメント作成の基本的な流れを紹介しました。  
ここでは、具体的な出力フォーマットの設定について紹介します。

## 4.1 Pandocについて



見えないのはわかっていますが…  
<https://pandoc.org/>

- **Pandoc**は、極めて多種のファイルフォーマットを相互に変換するツール
- R Markdownは、Pandocを利用して各種のフォーマットに出力している R Markdown → Markdown → 任意のフォーマット
- RStudioに同梱されているので、基本的にはインストール不要

23

R Markdownのさまざまなフォーマットへの出力は、大部分Pandoc ( <https://pandoc.org/> )というプログラムに依存しています。Pandocは多種多様な形式に対応した、文書ファイルのコンバーター (変換ツール) です。2.2節で述べたように、R Markdownファイルは、チャンクを実行した後、Markdownファイル (.md) に変換されます。そして、Markdownファイルを、Pandocを使ってさまざまな形式に変換しています。以降で紹介する、出力フォーマットごとのオプションも、大半はPandocに渡す設定です。

RStudioに最新のPandocが同梱されているため、別途インストールする必要はありません。

なお、2021年頃から、PandocをもとにしたQuartoという新しいドキュメント生成システムが利用されるようになり、RStudioもQuartoを同梱しています。それに伴い、**Pandocは、Quartoのインストールフォルダー以下に配置されています。**RStudioからのみR Markdownを利用する分には特に意識する必要はありませんが、他のエディターなどを使用する場合、RStudioを更新した際にPandocへのパスなどを修正する必要があります。

- 参考: Quarto <https://quarto.org/>



## 4.2 HTML



- output に `html_document` を指定すると、1ページのHTMLファイルとして出力できる
- 挿入した画像やプログラムで作成したグラフィックスもすべてデータ化（Base64エンコード）され、埋め込まれている
- ヘッダーで任意のCSSを指定できる
- （特に伝統的な大企業の）ビジネスにおいては、HTMLで報告、納品はあまり好まれない

24

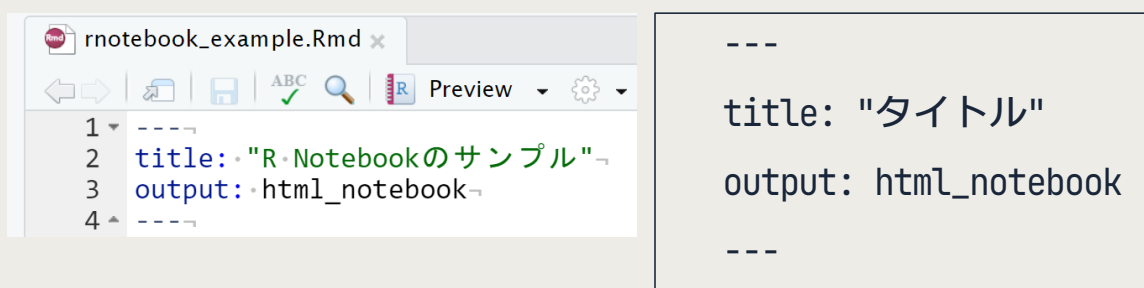
R Markdownの出力フォーマットのうち、最もスタンダードで、さまざまな記法、プログラムの出力に対応しているのが、HTMLです。ヘッダーに `output: html_document` と指定します。指定できる主なオプションに、以下のものがあります。

- `toc: true / false`: 目次を出力するかどうか
- `toc_depth`: 数値: 目次として表示する項目の深さ (1が#のみ)
- `toc_float: true / false`: 目次をスクロールに沿って移動するフローティングにするかどうか
- `number_sections: true / false`: 見出しに章・節番号を付けるかどうか
- `theme`: テーマ名: ページ全体のデザインをBootstrapテーマ集の中から指定する
- `highlight`: ハイライトテーマ名: プログラムの色付け (Syntax Highlighting) を `default`, `tango`, `pygments`, `kate`, `monochrome`, `espresso`, `zenburn`, `haddock`, `breezedark`, `textmate` から指定する
- `css`: `css`ファイル名: 独自のCSSを指定する
- `df_print`: データフレームの出力設定: データフレームを表示する際の設定を、`default`, `kable`, `tibble`, `paged` から指定する

CSSを駆使することで、自由にデザインを変更できます。しかし、R Markdownが出力するHTMLの構造を理解していないと、そもそもデザインできません。

## 4.3 R Notebook

- R Markdownの出力フォーマットのひとつで、ソース（.Rmd）を更新して保存するたびにRが実行され、結果がリアルタイムに反映される
- RStudioの "New File" から選択するか、output: html\_notebook と指定する
- RStudioの "Preview Notebook" メニューで出力を確認できる



25

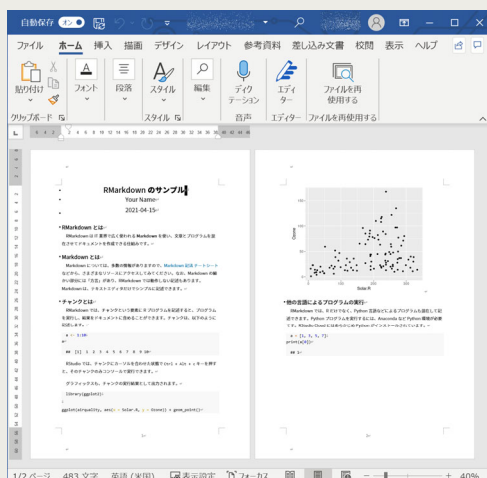
RStudioでは、R Markdownの他に、R Notebookという形式のドキュメントも作成できます。R Notebookは、HTML出力を拡張したもので、ドキュメント（本文、プログラム）を編集して保存するたびに、リアルタイムに表示が更新されます。

R Notebookを作成するには、RStudioのメニューから、"File" - "New File" - "R Notebook" と選択するか、ヘッダーに `output: html_notebook` と指定します。メニューから作成すると、自動的にテンプレートが表示されるので、内容を削除して、ドキュメントを作成します。

R Notebookでは、メニューに "Preview" ボタンが表示されます。これを押すことで、Viewerペインまたは別ウィンドウにR Notebookが表示されます。ソースファイルを編集し、保存すると、ドキュメントが更新されます。なお、プログラムの実行は自動では行われず、チャンクにカーソルを合わせて `Cmd + Enter` (行単位) か `Cmd + Shift + Enter` (チャンク単位) を押すと実行され、その後保存した際に出力が反映されます。

R Notebookの出力は、\*.nb.html というファイル名になります。

## 4.4 Word (.docx)



- output に word\_document を指定すると、Word形式 (.docx) で出力できる
- reference\_docx オプションに、あらかじめ用意したテンプレートファイル (.docx) を指定できる
- 見出しのサイズやフォント、行送りなど日本語文書としてそこそこ整った見栄えにするには、テンプレートを作りこむ必要がある

26

次に、Word形式の出力について紹介します。Markdownによる見出しやRプログラムの実行結果としての図表も含め、体裁の整ったドキュメントとして出力できます。基本は、output に word\_document または bookdown::word\_document2 と指定します。指定可能な主なオプションは、HTMLの場合と変わりませんが、reference\_docx オプションにテンプレートファイルを指定し、独自にカスタマイズしたデザインを反映させることができます。

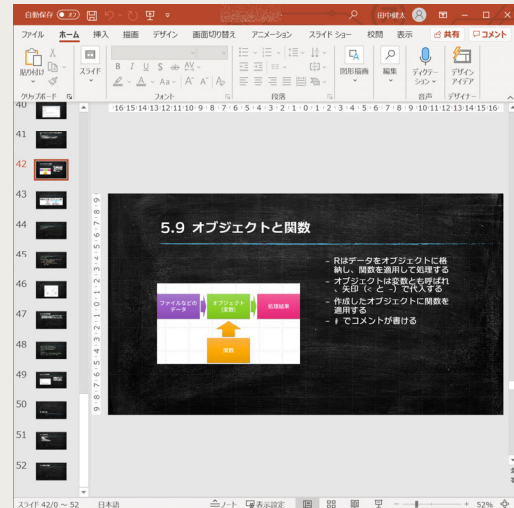
reference\_docx でテンプレートを指定する場合は、自分で用意する必要があります。まず、以下のコマンドを実行して、テンプレートファイルのひな型を出力し、自分好み（あるいは相手が指定した仕様）にカスタマイズしましょう。WindowsやmacOSで pandoc コマンドが見つからない場合は、Pandocがインストールされているディレクトリを探し、パスを指定する必要があります。

```
pandoc --print-default-data-file reference.docx > reference.docx
```

R Markdownから直接PDFを出力することもできますが、LaTeXの環境構築の手間や、見た目の「論文っぽさ」がビジネスにおいては敬遠されることも多いため、最終的な提出物がPDFであっても、一度Word形式で出力し、レイアウトを整えて、Wordの機能でPDFに変換するのが現実的ではないかと思います。

## 4.5 PowerPoint (.pptx)

- output に bookdown::powerpoint\_presentation2 や powerpoint\_presentation を指定すると、PowerPoint形式 (.pptx) で出力できる
- reference\_doc オプションに、あらかじめ用意したテンプレートファイル (.pptx) を指定できる
- 日本語スライドとしてそこそこ整った見栄えにするには、テンプレートを作りこむ必要がある



27

R Markdownから、PowerPointスライドも作成できます。実際に、この教材もR Markdownで作成しています。ITエンジニアの集まりではともかく、一般企業における「プレゼンテーション」はPowerPointで行うもの（Googleドキュメントでもなく）なので、自分以外の誰かがファイルを使用することが考えられる場合は、PowerPointで作成するのが良いでしょう。

PowerPointにおいても、HTMLと同様のオプションが指定できます。PowerPoint特有の項目は以下のようなものがあります。

- reference\_doc: "テンプレートファイル名": あらかじめカスタマイズしたテンプレートを指定する
- master: "スライドマスター名": テンプレート内の「スライドマスター」の名前を指定する
- slide\_level: レベル: 「節」スライドの開始レベルを指定する。章タイトルがある場合は2を指定する
- number\_sections: true / false: 見出しに章・節番号を付けるかどうか。2023年12月現在、PowerPointへの出力では、true にするとPandocのエラーが出て変換できない。そのうち解決するかもしれないが、現状は false として手で番号を振る必要がある

(次のページへ続く)

また、PowerPointでは2段組レイアウトと「ノート」が使用できます。R Markdownファイル内でそれぞれ以下のように記述します。

```
:::::{.columns}
```

```
 :::{.column width="48%"} # カラムの幅を指定
```

左カラムの内容

```
 :::
```

```
 :::{.column width="48%"} 
```

右カラムの内容

```
 :::
```

```
:::::
```

```
 ::: notes
```

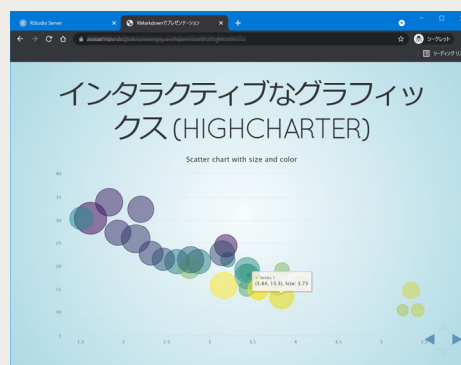
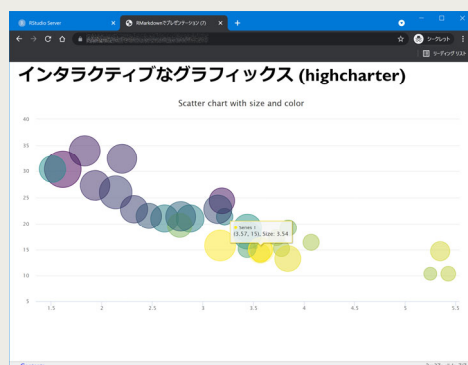
ノートに出力する内容

```
 :::
```

なお、2段組の幅については、レイアウトが崩れるので合計100%以下にする必要があるようです。

## 4.6 Webプレゼンテーション

- 近年、特に技術者コミュニティでHTML5でスライドを作成することが増えている
- R Markdownから、reveal.jsやslidyをはじめ、さまざまなWebプレゼンテーションを作成できる
- CSSを駆使した高度なスライド作成パッケージとしてxaringan（写輪眼）がある



29

逆 (?) に、ITエンジニアの集まり（技術部門の勉強会、OSSコミュニティなど）では、PowerPointを見かける機会は多くありません。近年は、HTMLを駆使したWebプレゼンテーションが使われることが増えています。Webプレゼンテーションは、HTML 5やJavaScriptを駆使した、ブラウザで操作できるスライドです。当然、Webでの公開と相性が良く、情報を広く共有するのに便利です。

Webプレゼンテーションのための (JavaScript) ライブラリは多数ありますが、R Markdownでは、広く使われているreveal.jsやslidyをサポートしています。また、CSSについて熟知したユーザー向けの高度なパッケージとして、xaringan（写輪眼）があります。

Webプレゼンテーションについては、使用するライブラリやオプションが多岐にわたるので、ここでは詳しく触れません。以下の記事などを参考にしてください。

- reveal.js: <https://revealjs.com/>
- slidy: <https://www.w3.org/Talks/Tools/Slidy2/>
- xaringan: <https://github.com/yihui/xaringan>



## 4.7 PDF (LaTeX使用)

- 論文を執筆するためのツールとして、伝統的にLaTeXが使われる
- R MarkdownではLaTeXの記法で数式を記述できる
- R MarkdownからLaTeX形式に変換し、PDFを作成できる
- 設定次第で、一般的なA4縦のフォーマットも、Beamerによるスライドも作成できる
- Rとは別にLaTeX環境を整える必要があり、手間がかかる
- ビジネスの場では、LaTeXを使う機会は、残念ながらほぼない

30

研究の世界では、伝統的に論文執筆のために、LaTeX (らてふ) という組版システムが使われてきました。LaTeXは、優れた組版 (レイアウト) 機能と高度な数式出力機能で、理工系論文を執筆する際には欠かせません。しかし、Rなどのプログラミング環境とは独立しているため、前述のような分析結果の更新漏れ、といったことが起こり得ます。

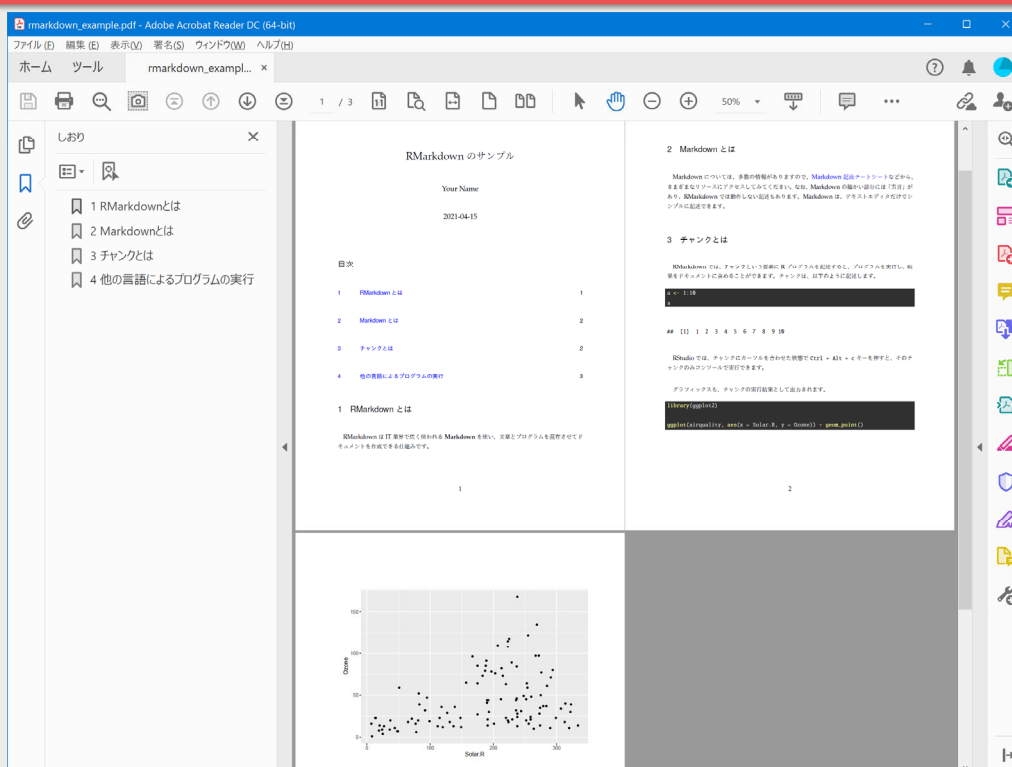
R Markdownでは、MarkdownとR言語などで作成したドキュメントを、LaTeX形式に変換し、さらにLaTeX (関連ソフトウェア) の機能で、PDFに変換できます。ただ、LaTeXへの出力をするためには、RやRStudioとは別に、LaTeX環境を整備する必要があります。本格的にLaTeXを使うのであれば、TeX Liveというディストリビューション (配布物) を導入します。詳細は、以下の記事などを参考にしてください。

- 参考: TeX入手法 - TeX Wiki  
<https://texwiki.texjp.org/?TeX%E5%85%A5%E6%89%8B%E6%B3%95>

一方で、R MarkdownからしかLaTeXを使わない、という場合はよりコンパクトな、tinytexパッケージを導入することもできます。`install.packages("tinytex")` とすることで、日本語にも対応した最小限のLaTeX環境が (R専用として) 導入されます。

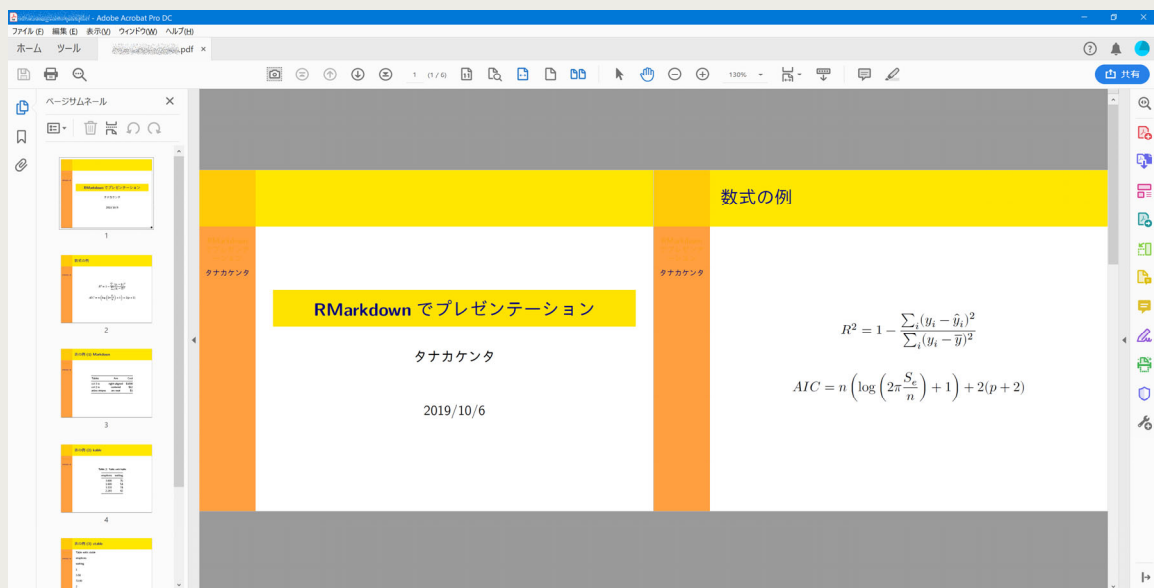
実際のところ、ビジネスにおいて (研究職以外で) LaTeXを使う機会はほとんどありません。また、R Markdownにおける設定も、他の出力フォーマットよりも複雑です。そのため、ここでは詳しく触れません。

## 4.7 PDF (LaTeX使用)





## 4.7 PDF (LaTeX使用)



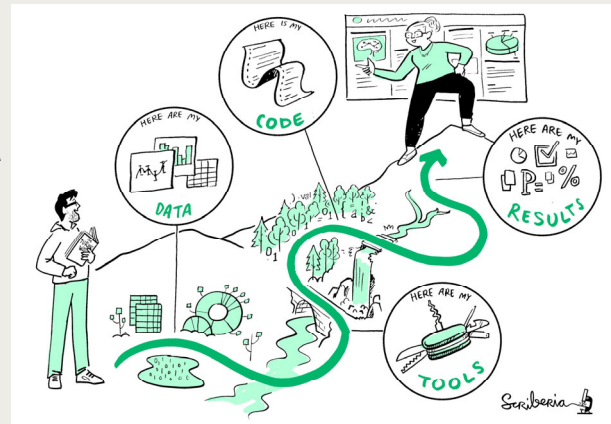
## 5. Reproducible Researchのために

33

ここからは、R Markdownそのものではなく、RやR Markdownを使った研究活動を行う上で重要な、Reproducible Researchという考え方を紹介します。

## 5.1 Reproducible Researchとは

- Reproducible Researchは、「**再現性のある研究**」というように訳される
- **自分の研究（分析）を、第三者が再現できるように管理しよう**という考え方
- 他人でなくても、自分が後から成果を再利用できるよう、わかりやすい構造でプログラムやデータを管理することが求められる



<https://the-turing-way.netlify.app/reproducible-research/reproducible-research.html>

34

研究の世界ではReproducible Researchという考え方が重視されています。直訳すると、再現性のある研究、といったようになります。近年、研究成果の捏造や特許などにまつわる知財紛争などで、「誰が、いつ、どのようにして成果を出したのか。同じ手順で作業すれば同じ結果が再現できるか」という、研究プロセスを記録することの重要性が広く知られるようになりました。(もちろん、それ以前から「科学」は再現性の追求が一つのテーマではありますが)

収集したデータや、その分析結果、そして作成したドキュメント（研究日誌、進捗報告資料、論文など）をきちんと管理しなければ、自分自身でさらに研究を進める際や、共同研究を持ち掛けられた時、そして研究成果に疑いを持たれたときに、適切なデータを参照することができません。ビジネスにおいても、意思決定には根拠が必要で、意図しない結果になった時には再現性の検証が必要です。その意味で、Reproducible Researchを意識することは、データ分析を行う上で必須の態度と言えるでしょう。

RStudioそしてR Markdownは、このようなReproducible Researchをサポートする機能が充実していますので、これから紹介していきます。

## 5.2 プロジェクトのセットアップ

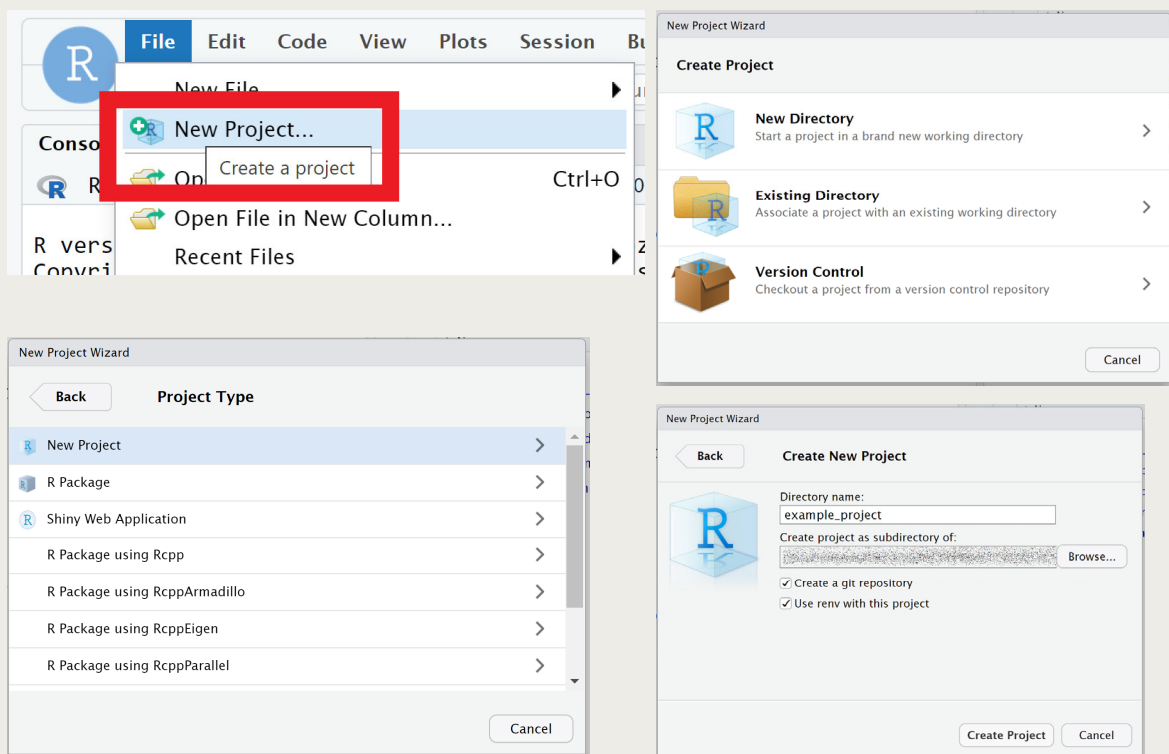
- RStudioでは、プロジェクトを作成し、関連するファイルを管理できる
- "File" - "New Project" を選択し、プロジェクトを作成する
- "Use renv…" にチェックを入れると、パッケージのプロジェクト単位のインストールやバージョン管理もできる
- RStudio Cloudでは自動的にプロジェクトを作成した状態から始まる

35

RStudioでは、さまざまなフォルダーに存在するファイルを個別に編集することもできますが、プロジェクト機能を使い、プログラムやデータ、ドキュメントを一元管理できます。それほど複雑な機能ではなく、任意のフォルダー以下にプロジェクト（研究、授業、課題などさまざまな単位）のフォルダーを作成し、その中にプロジェクトに関連するプログラムやデータ、ドキュメントを格納する、というものです。プロジェクトを作成すると、指定したフォルダーにプロジェクト名.rproj というファイルが作成されます。以降は、WindowsやmacOSのファイルエクスプローラーで、.proj ファイルをダブルクリックすれば、RStudioが起動し、自動的にプロジェクトが開きます。

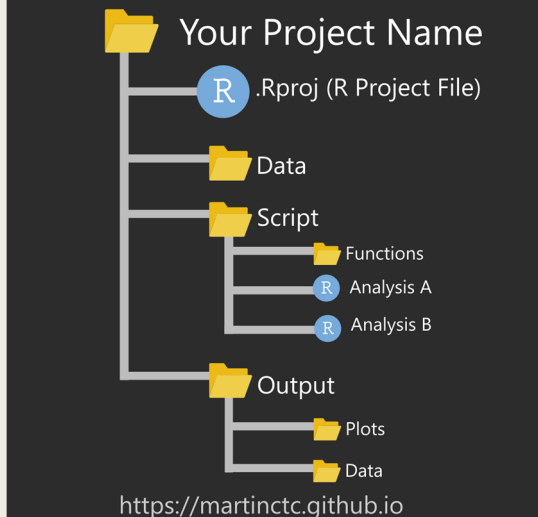
また、高度な設定として、プロジェクトを作成する操作の中で、"Use renv…" というオプションがあります。これにチェックを入れると、renvというパッケージの機能を使い、プロジェクト固有のパッケージリポジトリが作成されます。研究やレポートに必要なパッケージをインストールする際に、依存関係など影響が大きく、全体の環境には導入したくない、といった時に、renvを使うと、プロジェクトにだけ有効な形でインストールできます。また、事情があってプロジェクトでは古いバージョンのパッケージを使わなければならない時などにも便利です。

## 5.2 プロジェクトのセットアップ



## 5.3 プロジェクトのディレクトリ構造

### A basic R project set up



- 正解はないが、ファイルの種類、作業の目的ごとにディレクトリ（フォルダー）を分けて管理する

- あらかじめ空のフォルダー構造を作っておいて、作業を開始する際にコピーして使うのが手軽

<https://martinctc.github.io/blog/rstudio-projects-and-working-directories-a-beginner%27s-guide/>

37

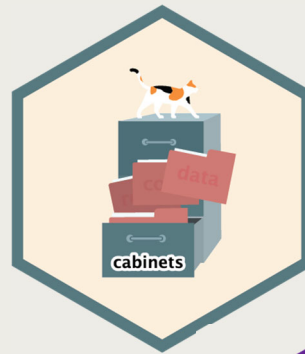
RStudioのプロジェクト機能を使っても、プロジェクトフォルダーの中でデータやプログラムがばらばらに配置されていると、管理がしにくいです。プロジェクトフォルダーの下に、一貫したポリシーのもと、目的別にサブフォルダーを作成し、ファイルを分別して格納する必要があります。

どのようなフォルダー構造にすべきか、ということに正解や標準はありませんが、データサイエンスの実務に当たる人達の、経験に基づく提案がされています。以下の記事やパッケージなどを参考にしてください。

- 研究プロジェクトの管理：フォルダ構成とGitHubのミニマムな利用  
[https://qiita.com/Shuntaro\\_Sato/items/2d4859d732596729d28f](https://qiita.com/Shuntaro_Sato/items/2d4859d732596729d28f)
- RstudioをGitHubとプロジェクトテンプレートで管理する  
<https://plaza.umin.ac.jp/shoei05/wp/index.php/2020/10/25/165/>
- RStudio Projects and Working Directories: A Beginner's Guide  
<https://martinctc.github.io/blog/rstudio-projects-and-working-directories-a-beginner%27s-guide/>

## 5.4 パッケージを活用した環境構築

- 作業ディレクトリの作成やバージョン管理を統合的に行うことができるパッケージも複数存在する
- cabinetsパッケージやworkflowrパッケージが広く使われている様子
- 最近は、Git (GitHub) によるバージョン管理が当たり前になっている



38

前節で紹介したように、プロジェクト配下のフォルダー構造は、自分が使いやすいように、一貫した方針で作成すると良いでしょう。一方で、そのようなテンプレートを自動的に作成したい、というニーズも多くあるため、Rプロジェクトの典型的な構成を行うためのパッケージとして、以下のようなものが提供されています。

- ProjectTemplateパッケージ <http://projecttemplate.net/index.html>
- cabinetsパッケージ <https://github.com/nt-williams/cabinets>
- usethisパッケージ <https://usethis.r-lib.org/>

また、手動で作成するか、上記のようなパッケージを活用するかによらず、現在では、プロジェクトをGitHubでバージョン管理することが一般的です。GitHubの使い方は、Rに限定したものではないのでここでは詳しくは触れませんが、以下のガイド記事などを参考にしてください。

なお、GitHubでは、設定次第で作成したリポジトリを全世界に公開することになります。公開することで研究内容が公知になってしまったり、リポジトリに他者の著作物を含めたことで知財を侵害したり、さまざまなリスクがありますので、GitHubの公開設定には十分注意してください。

- 参考: 【2021年版】Git入門者が1日目に覚えるべき操作まとめ  
<https://qiita.com/enhrs/items/8c87acbafc4747ff856b>

## 6. まとめ



## 6.1 今日の内容

---

- R Markdownによるドキュメント作成
  - R Markdownの概要
  - Markdownの基礎
  - R Markdownのファイル構造
  - さまざまな出力フォーマット
- プロジェクトによるReproducible Research
  - Reproducible Researchとは
  - RStudioのプロジェクト機能
  - パッケージを活用したプロジェクト管理

## 6.2 次回までの課題

---

- RStudio Cloudプロジェクト内の  
03\_rmarkdown\_exercise.Rmd について、  
指示に従ってプログラムを作成してください
- 編集したファイルは、ファイル一覧でチェックを入れ、  
[more] メニューから [Export] を選択し、  
[Download] ボタンを押してダウンロード  
してください
- 今回の提出物は Rmd ファイルと docx ファイルの  
2つです
- ダウンロードしたファイルを、Classroomの  
課題ページから提出してください
- 提出期限: 2023-10-16 23:59