

R Markdown で日本語 beamer プレゼンテーション

ill-identified

2020-07-04

イントロダクション

このスライドは何？

- あまり情報が流れていない, rmarkdown と beamer で日本語を含むスライドを作るためのテンプレート兼用例集
- reveal.js など html 媒体は他の資料を参照
 - [ここ](#)や[ここ](#)を見よ
- もともとは自分用に作ったテンプレだったものを万人向けに修正

想定される用途

- Tokyo.R など R を使った話を発表する際の資料作成
- 技術・アカデミック寄りの話題を想定
- 具体的に要求されるもの
 - **日本語表示**
 - ラスタまたはベクタ画像の挿入
 - 表の挿入
 - R コードを見やすく表示
 - 参考文献の相互参照/リスト自動生成
 - **LyX や overleaf より簡単であること**
 - **なんかナウでオサレな感じは求めてない**
 - 自由すぎるデザインは不可

先行事例の紹介

- 伊東『R Markdown と Beamer でプレゼンテーション資料作成』
 - Lua \LaTeX を使って日本語で Beamer スライド作成する方法
- Atusy『R Markdown + XeLaTeX で日本語含め好きなフォントを使って PDF を出力する』
- 先行事例との違い:
 - エンジンを Xe \LaTeX に変更
 - 日本語文献 bib ファイル・bst ファイルに対応
 - スライド作例を多少充実させた
 - その他体裁にこだわりたい人向け
 - 「表 X」「図 X」といったキャプション

reveal.js じゃダメなの？

- 個人的にデザインとかあまり好きじゃない
- 上下左右に動いて空間識失調になる
 - (個人の体験です)
 - 上下のみにもできる
- html よりも不変な媒体にしたい
 - pdf が明確に優れているかは怪しい
- Q: お前が使いこなせてないだけじゃないの?
 - A: うるさい

パワーポイントじゃダメなの？

- 私は**持っていない**
- シンタックスハイライトが面倒
 - パワポの場合は**VSCode**か**reprex**でコピペ
- ドラッグ & ドロップで位置調整は便利
- しかしポンチ絵芸術になりがち
- 極力シンプルにして視線誘導の負担をなくすべき
 - 徹底するかは**好みの問題**

技術的に厄介だったところ

- html と pdf(L^AT_EX) とで微妙に違う挙動
 - ネット上の情報は html 前提が多い
 - pandoc チョットワカル必要
- 日本語を含む参考文献リスト
 - upBibT_EX の適用
 - 細かいオプション, 特に metropolis 特有の仕様
- RStudio Cloud で動くかは未確認
 - 日本語表示がおかしい説あり

使い方

セットアップ

1. パッケージのインストール

```
remotes::install_github(  
  "Gedevan-Aleksizde/my_latex_templates",  
  subdir = "rmdCJK")
```

2. TeXLive (>= 2018) のインストール

- 分からなければ [TeX wiki のページ](#) を参考に

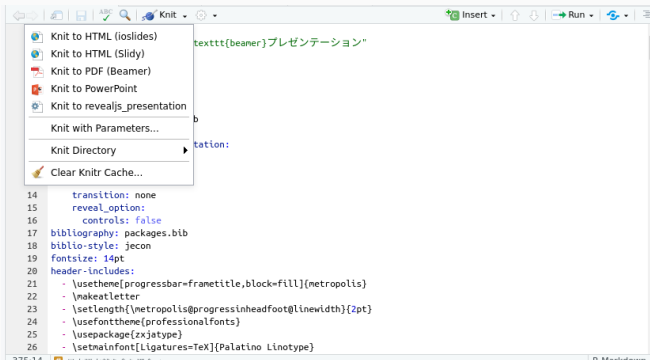
基本

1. yaml ヘッダに以下を書く

- 詳しくは example/ のテンプレートを参考に

output: rmdCJK::beamer_presentation_CJK

2. RStudio のツールバーの “Knit” を押す



フォント指定

- 使うマシンに応じて以下の箇所を適当に変える
 - 参考: [Google Fonts](#)でフリーフォント入手可能
 - インラインでのフォント変更は面倒 (後述)

```
mainfont: Roboto  
sansfont: Noto Sans CJK JP  
monofont: Ricty Diminished  
CJKmainfont: Noto Sans CJK JP
```

基本構文

- markdown 的な書き方でできる
- “## タイトル” でスライドの開始
 - \LaTeX コマンドも挿入可能

```
# 節見出し
```

```
## タイトル 1
```

```
- ** 太字 ** **bold**
```

```
- _ 強調 _ _emph_
```

```
- `タイプライタ体` `mono`
```

- **太字** **bold**
- 強調 *emph*
- タイプライタ体 `mono`

Beamer や RMarkdown 使用に役立つ資料

- 伊東『R Markdown と Beamer でプレゼンテーション資料作成』(Lua \LaTeX 使用)
- 松田『Beamer 読本-講演用スライド作成のために-』
- Kazutan『R Markdown によるスライド生成』『R Markdown 入門』
- Atusy『R Markdown + XeLaTeX で日本語含め好きなフォントを使って PDF を出力する』
- R Markdown 2.0 チートシートの日本語訳, Takahashi, M. 訳

もう少しくわしいやつ

- Atusy 『R Markdown ユーザーのための Pandoc's Markdown』
- 謝益輝 (yihui) “knitr - Elegant, flexible, and fast dynamic report generation with R” (開発者本人)
- Xie, Yihui & C. Dervieux “R Markdown Cookbook”

今回使うパッケージ

- このファイル作成には以下を使用している
 - 図表作成とか最低限必要なものだけ

```
01 require(conflicted) # パッケージの競合防止用
02 require(tidyverse)  # 全般
03 require(ggthemes)   # ggplot2 のデザイン変更
04 require(ggraph)     # ネットワーク図の用例に
```

- 以下はインストールのみ/読み込む必要なし
 - citr: 引用文献の挿入を GUI で
 - bookdown: 数式を GUI で

ソースコードの表示: 基本事項

- `echo=T` でチャンク内コードを表示
 - デフォでは非表示
 - **自動でシンタックスハイライト**
- はみ出す場合は `tidy=F` して手動改行
 - 日本語等で折り返し地点がうまく行かない
- `class.source = "numberLines, LineAnchors"` で行番号表示 (参考)

ソースコードの表示: 出力例

```
```{r, echo=T, class.source = "numberLines, LineAnchors"}  
require(conflicted)
require(tidyverse)
require(ggthemes)
require(ggdag)
```
```

```
01 require(conflicted)  
02 require(tidyverse)  
03 require(ggthemes)  
04 require(ggdag)
```

数式関係

数式の挿入: 行内 (インライン)

- markdown 風の LaTeX コード埋め込み
- \LaTeX の数式を $\$$ で挟む
- 例: らんま $\frac{1}{2}$
 - 出力: らんま $\frac{1}{2}$
 - 注: 行内で分数はスラッシュ使ったほうが見やすい
- 数式にはセリフフォント使用
 - スライドはサンセリフが良いとされる
 - しかし数式の統一感がない
 - (個人の好み?)

数式の挿入: 独立行

- \$\$ で挟んだ範囲に \LaTeX 構文

```
$$\begin{aligned}& \sin^2(x) + \cos^2(x) = 1 \\& f(x) = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^n} \hat{f}(\omega) \exp(i\omega x) d\omega \\& \end{aligned}$$
```

$$\sin^2(x) + \cos^2(x) = 1$$

$$f(x) = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^n} \hat{f}(\omega) \exp(i\omega x) d\omega$$

数式の挿入: bookdown パッケージのアドインで補完

1. RStudio のツールバー “Addins”
2. “Input LaTeX Math”

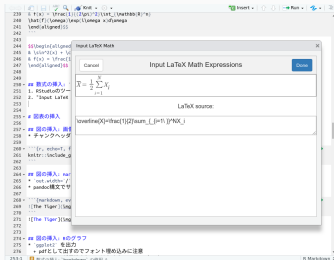


図 1: bookdown の数式入力機能

- 一部対応してない記号もある?
 - `\mathbb{b}{}` とか `\hat{}` とか
- 数式のみで `\aligned` 等環境の入力は不可

図表の挿入

図の挿入: 画像ファイル貼り付け

- チャンクの `out.width=/out.height=` で調整
- html と違い**アスペクト比は固定**
- jpeg, png, eps, pdf に対応
 - gif, svg は上記いずれかに**手動で変換**する必要
 - \LaTeX (\XeLaTeX) の制約

```
knitr::include_graphics(file.path(file_loc, c("img/tiger.eps
```



図 2: いつもの虎 (TeXLive より)

図の挿入: markdown 構文で貼り付け

- `out.width=/out.height=` が適用されない
- pandoc 構文でサイズ指定

```
![The Tiger](img/tiger.pdf){ height=30% }
```



図 3: The Tiger

図の挿入: ggplot2 のグラフ

- `fig.cap=` でキャプションを設定可能. `labs(title =)` と違い自動相互参照あり

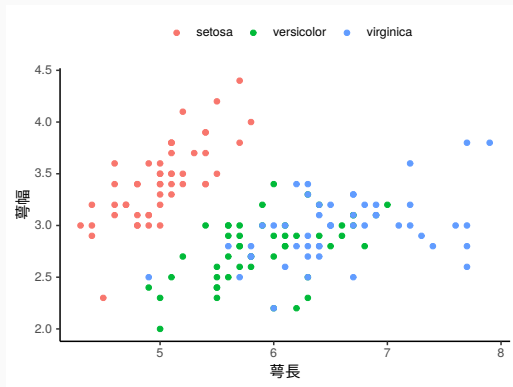


図 4: ggplot2 の出力例: iris データ

図の挿入: 文字の大きさをそろえるには

- RStudio と出力された画像ファイルが違う!
- グラフの文字小さすぎ!!
- その原因は

1. 自動縮小されるため

- 込み入った話なので**次のスライドへ**

2. 単位が違うため

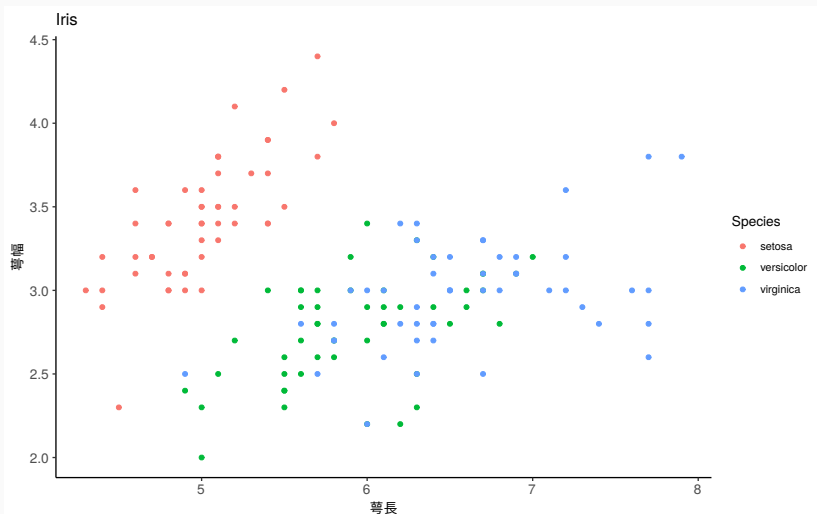
- beamer は主に **pt** 単位
- ggplot2 は `annotate()` のみ **mm** 単位
- 補足
 - `cairo_pdf()` の `pointsize` はビルトインデバイスにのみ影響
 - 『**ggplot2 の size が意味するもの**』

図の挿入: 画像サイズの基本ルール

- R が作図したファイルを一旦保存し, 拡大縮小して貼り付けられる
 - `fig.width/fig.height` は**保存時**のサイズ
 - `out.width/out.height` は**表示する**サイズ
- R の保存サイズと beamer スライドのサイズのデフォルトは違う
 - スライドは **5.04 x 3.78 in (128 x 96 mm)(4:3)**
 - `ggsave()` は **9.11 x 5.77 in** で保存
- RStudio のビューアは文字の大きさ**固定**で**サイズを画面に合わせる**
 - **違和感の正体 (?)**

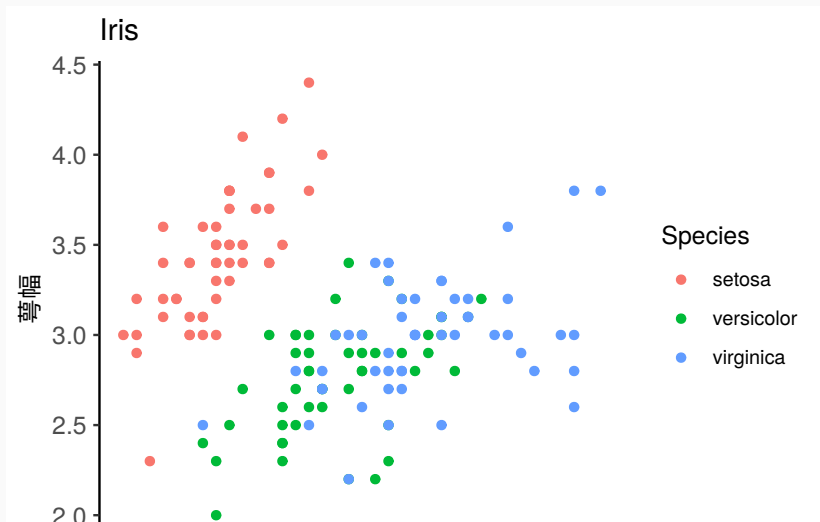
図の挿入: 幅 100% で出力

- 注: `out.width="100%"` はスライドサイズではなく **本文領域の相対サイズ**



図の挿入: beamer サイズで保存, 幅 100% で出力

- 相対的に文字が大きくなった



図の挿入: 字の大きさをなるべく揃える

- 基準を beamer に合わせる方法
 1. 保存時サイズを beamer の画面サイズと同じにする
 - このテンプレートのデフォルト設定
 2. `theme_*`() で `base_size` を beamer の文字サイズと同じにする
- `out.width="100%"` のとき, グラフタイトルと本文のサイズが一致
- 拡大縮小に合わせて文字の大きさを調整する
- 横長のグラフなら `fig.width=` を調整する

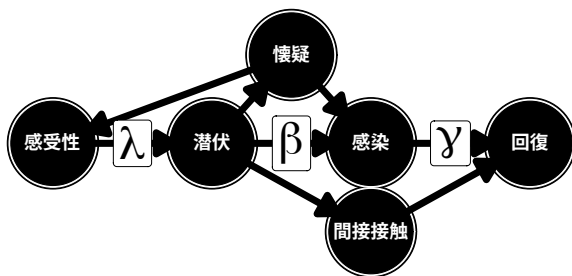
図の挿入: 再現可能なポンチ絵

- 概念図とかの図示はどうか
 - NOT データの視覚化 (ビジュアライゼーション)
 - ggplot2 の本来の使い方ではない
- ggdag はネットワーク図に使える
 - 因果ダイアグラム, 遷移図, グラフィカルモデル等
- ggforce はベン図の描画に応用可能
 - 世間的にはグラフの部分拡大用パッケージ?
- 詳しくは個別のマニュアル参照
- 霞が関流ポンチ絵は**専門外**

図の挿入: ポンチ絵の例 1

- 以前作ったやつの修正

Warning: Removed 6 rows containing missing values (geom_label



図の挿入: ポンチ絵の例 2

- `ggforce::geom_circle()` を利用
 - 参考: [How to Plot Venn Diagrams Using R, ggplot2 and ggforce](#)

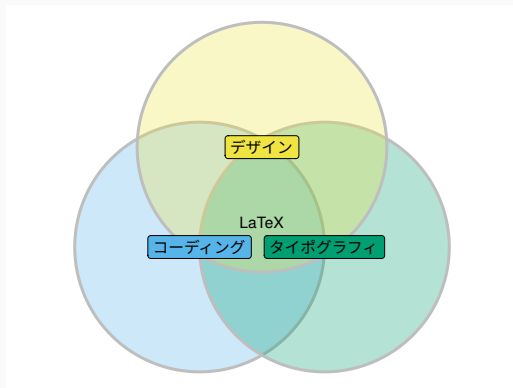


図 6: ベン図の例

図の挿入: R 以外のデバイス

- \LaTeX の tikz を使用可能
 - tikz を知らない人は[ここ](#)や[TeX Wiki](#)を読む
 - 現時点では**日本語表示が面倒** ([参考](#))
 - そこまでやるなら全部 ~~\LaTeX~~ で書いたほうがいいのか?
- dot 言語単体で実行することも可能

表の挿入: データフレーム

- R のデータフレームとして作成して出す
 - はみ出す場合は縮小
 - 最低限の情報だけ掲載するのは大前提
 - df_print: kable では caption 指定がややこしい

```
data(iris)
knitr::kable(head(iris[, 1:3]),
               caption="kable() による表示")
```

表の挿入: データフレームを `kable()` で表示

```
data(iris)
knitr::kable(head(iris[, 1:3]),
               caption="kable() による表示")
```

表 1: kable() による表示

| Sepal.Length | Sepal.Width | Petal.Length |
|--------------|-------------|--------------|
| 5.1 | 3.5 | 1.4 |
| 4.9 | 3.0 | 1.4 |
| 4.7 | 3.2 | 1.3 |
| 4.6 | 3.1 | 1.5 |
| 5.0 | 3.6 | 1.4 |
| 5.4 | 3.9 | 1.7 |

表の挿入: L^AT_EX コード

- L^AT_EX のコードを貼り付けて表を掲載
 - `\input{tab.tex}` でコピペなしで貼り付け可
 - stargazer との併用
 - **リサイズは手動で**
- 以下, 表を .tex で出力してから読み込む

```
xtable::xtable(  
  head(iris), caption = "xtable で export") %>%  
  print(file = "tab.tex")
```

| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | |
|---|--------------|-------------|--------------|-------------------|---|
| 1 | 5.10 | 3.50 | 1.40 | 0.20 | s |
| 2 | 4.90 | 3.00 | 1.40 | 0.20 | s |
| 3 | 4.70 | 3.20 | 1.30 | 0.20 ⁶ | s |

表の挿入: markdown 構文

Table: 得点一覧


| クラス | 科目 | 平均 |
|-----|----|--------|
| A | 算数 | \$90\$ |
| B | 算数 | \$95\$ |

表 3: 得点一覧

| クラス | 科目 | 平均 |
|-----|----|----|
| A | 算数 | 90 |
| B | 算数 | 95 |

外部資料の引用方法

ハイパーリンクの挿入

- url は自動でリンク
 - <https://rstudio.com/>
- markdown 方式のリンク
 - [RStudio](https://rstudio.com/)
 - [RStudio](#)
- 画像にハイパーリンク  RStudio[®] を貼ることも可

文献引用の方法

- `[@ref]` で番号引用: `\citep{ref}` に対応 ([1])
- `@ref` で著者名引用: `\citet{ref}` に対応 (hogehoge et al.)
- `[@ref1; @ref1]` で連番引用 [1, 2]
- 以下引用テスト

```
[@R-base; @R-bookdown; @R-citr; @wickham2016Data]
```

[? ? ? ?]

文献引用の補助: 引用子の補完

- 重複・書き間違えの防止
- citr パッケージを使うと楽
 - ツールバーの Addins から選択
 - zotero 連携機能あり

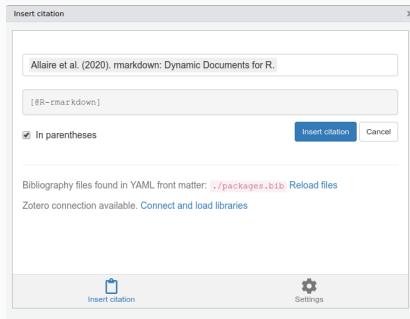


図 7: citr パッケージの GUI

文献引用の補助: 文献管理

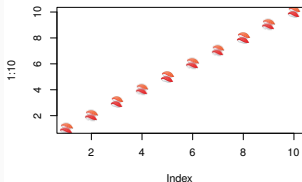
- Mendeley, Zotero, ReabCube の 3 つが多い?
- 私は Zotero を使っている
 - 多言語対応, 連携機能の充実, 料金などの理由
 - 参考: 『Mendeley Exodus Mendeley から Zotero への移行の手引き 〜』
- RefManageR パッケージ
 - R で bib ファイルをパースしたりする
 - 文献管理用には既存ソフトで十分?

その他の機能

絵文字

- **BXcoloremoji**をインストールすれば可能
 - `\coloremoji{}` で絵文字表示: 🍷
- グラフ描画には特に設定必要なし
 - ソースコード上のは文字化けする

```
plot(1:10, pch = "🍷")
```



基本的なカスタマイズ

フォント変更

- XeCJK パッケージで制御している
 - 本当は `zxjatype` でやりたかった
- `mainfont` は**セリフフォント**を指定する必要
 - スライドでもサンセリフを使ってしまうため

```
mainfont: Roboto
sansfont: Roboto
monofont: Ricty Diminished
CJKmainfont: Noto Sans CJK JP
```