



統計学 2

4. R Markdown による レポート作成

やない ゆうき
矢内 勇生



<https://yukiyanai.github.io>



yanai.yuki@kochi-tech.ac.jp



このトピックの目標

- R Markdownを使ってレポートを作成できるようになる
 - ▶ R Markdown ファイルを作る
 - ファイルの作成・保存
 - ▶ R Markdown ファイルを編集する
 - YAML ヘッダ
 - 文章
 - コードチャンク
 - ▶ R Markdown ファイルを knit する
 - HTML へ
 - PDF へ

R Markdown とは？

Markdownとは？

- 軽量マークアップ言語の1つ
 - ▶ テキスト形式で構造化された文書が書ける
 - 簡単な文書は markdown で書くのがおすすめ
 - ▶ スライドも作れる
- ファイル名拡張子は .md
- 自分の好きなテキストエディタ（例：VS Code）で編集できる
- 詳しくは、日本語Markdownユーザー会のウェブサイト <https://www.markdown.jp/what-is-markdown/> を参照

Markdownの例

- Visual Studio Code の場合

▶ 左半分が入力画面、右半分が出力（のプレビュー）



R Markdown とは？

- 簡単に言うと、Markdown にRの機能を追加したもの
- ファイル名拡張子は .Rmd
- Markdown の書き方がそのまま使える
- Rのコードも書ける
- Rの実行結果を文書の中に取り込める
 - ▶ 計算結果を書き写したりコピーしたりする必要がない
 - ▶ 図表もRで作成できるので、レポート用に作り直す必要がない（はじめからレポートに適した図表を作る）

RスクリプトとR Markdownの違い

Rスクリプト

- 拡張子：.R
- 基本はRコード
 - ▶ # でコメントを区別
- 目的：分析すること

R Markdown

- 拡張子：.Rmd
- 基本は文章
 - ▶ Rコードは**コードチャ
ンク**に書く
- 目的：レポートをまとめること

R Markdown の構成要素

- ヘッダ：YAMLでファイルの情報を書く
- 文章：通常の文章を入力する
 - ▶ 基本は markdown
 - ▶ HTML や LaTeXの数式も書ける
- Rコード：コードチャンクの中に書く
 - ▶ いつも通りRのコードを書けばよい
 - ▶ 稀に思い通り動かないコードがあるので、オプションでコントロールする

なぜ R Markdown を
使うのか？

分析結果を再生・再現できるようにする

- 今日実行した分析と同じ分析を、明日も明後日も・・・
1年後も・・・10年後も・・・できるようにする
- 近年は、研究成果を発表する際にデータと分析に使ったコードを同時に公開することが求められることが増えてきた
 - ▶ 研究の透明性を確保する
 - ▶ 互いの研究をより深く理解する

分析の過程を記録する

- 研究・分析過程を細かく記録する
 - ▶ 分析の結果だけでなく、過程全体を記録する
- 成果を発表するときは、分析に使ったRコードも公表する
 - ▶ 他人が読んでも理解できるように読み易いコードを書かなければいけない
 - ▶ 他人が読んでもわかるように、コードの説明をしなければいけない
 - 「何をしたか」はコードからわかることが多い
 - 「なぜそれをしたか」はコードからわからないので、詳細に説明する

分析過程を記録するメリット

- コードの再利用が可能
 - ▶ コンピュータの電源を落としても、途中から分析を再開することが可能
 - ▶ 過去のプロジェクトで使ったコードの再利用が可能に
- 分析手続きの洗練化
 - ▶ 分析手続きの見直しが容易に
- 他人のためだけでなく、未来の自分のために

R Markdownのメリット

- 文章とRコードの両者を扱える
- 説明と分析が1つのファイルで完結する
 - ▶ 情報の拡散が防げる
 - ▶ すべて記録できる
 - レポートで「見せる」部分と「見せない」 (隠蔽するとい
う意味ではなく、ファイルには残すが画面に出力させない部分) 部分を選
べる
- RStudio のプロジェクト機能で管理可能

R Markdown ファイルの 作り方

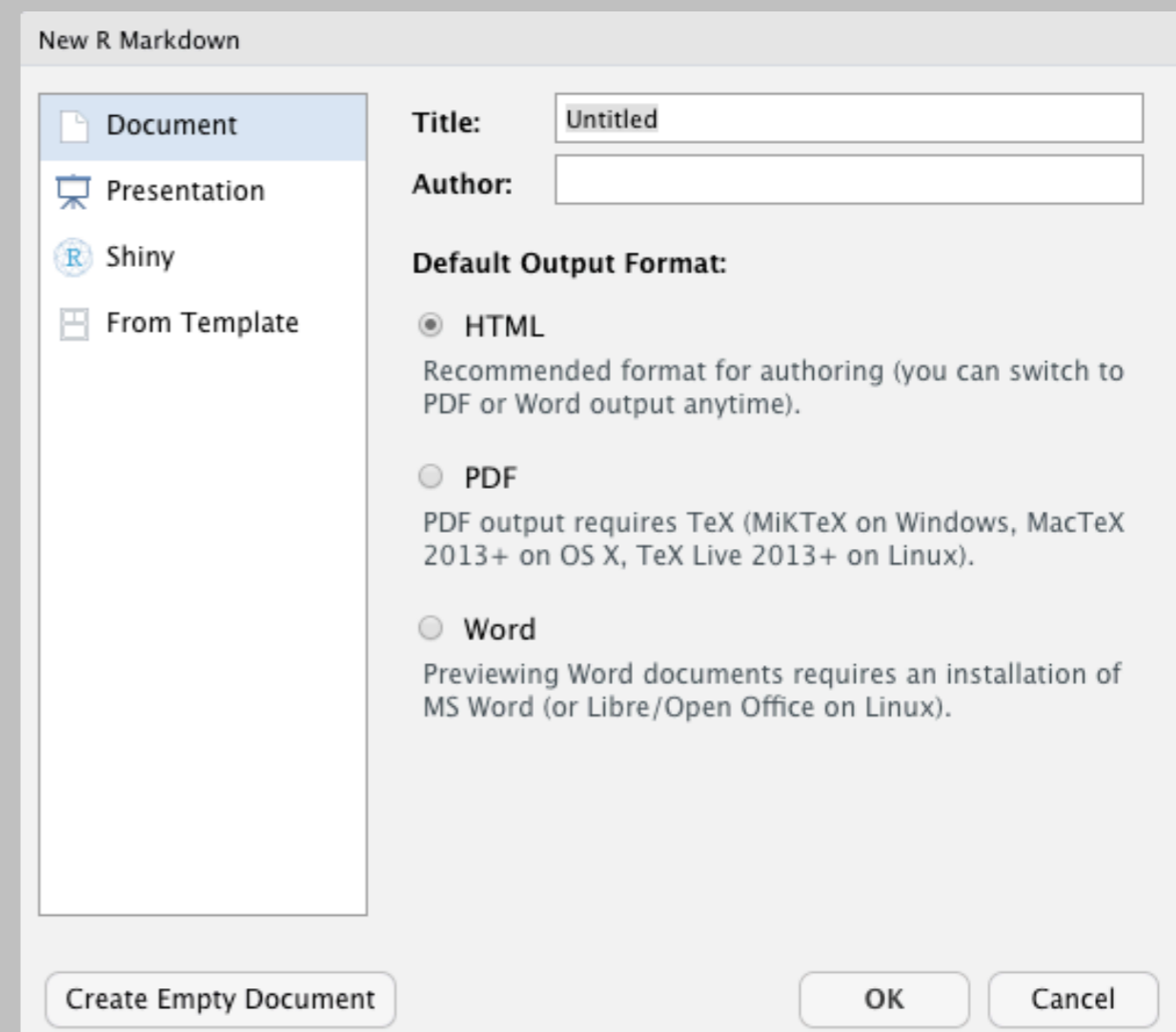
R Markdownファイルの作成 (1)

- RStudio の左上のファイル作成ボタンをクリックし、[R Markdown...] を選ぶ



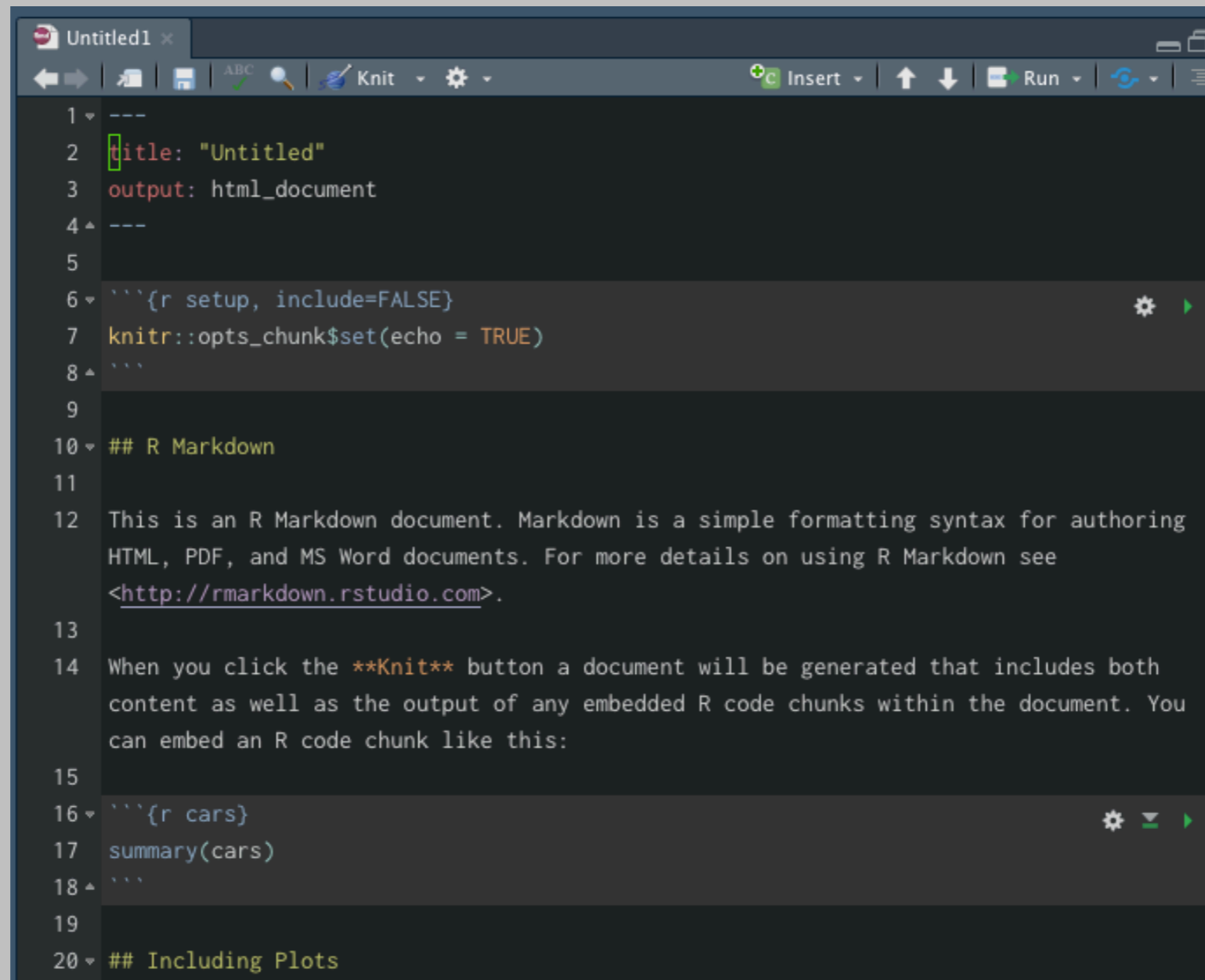
- ▶ 初めて開くときに追加で必要なパッケージのインストールを促される（英語でメッセージが出る）ので、Yes [OK] をクリックする（ネット接続が必要）

- 右のような画面が出たら、[OK] をクリックする



R Markdown ファイルの作成 (2)

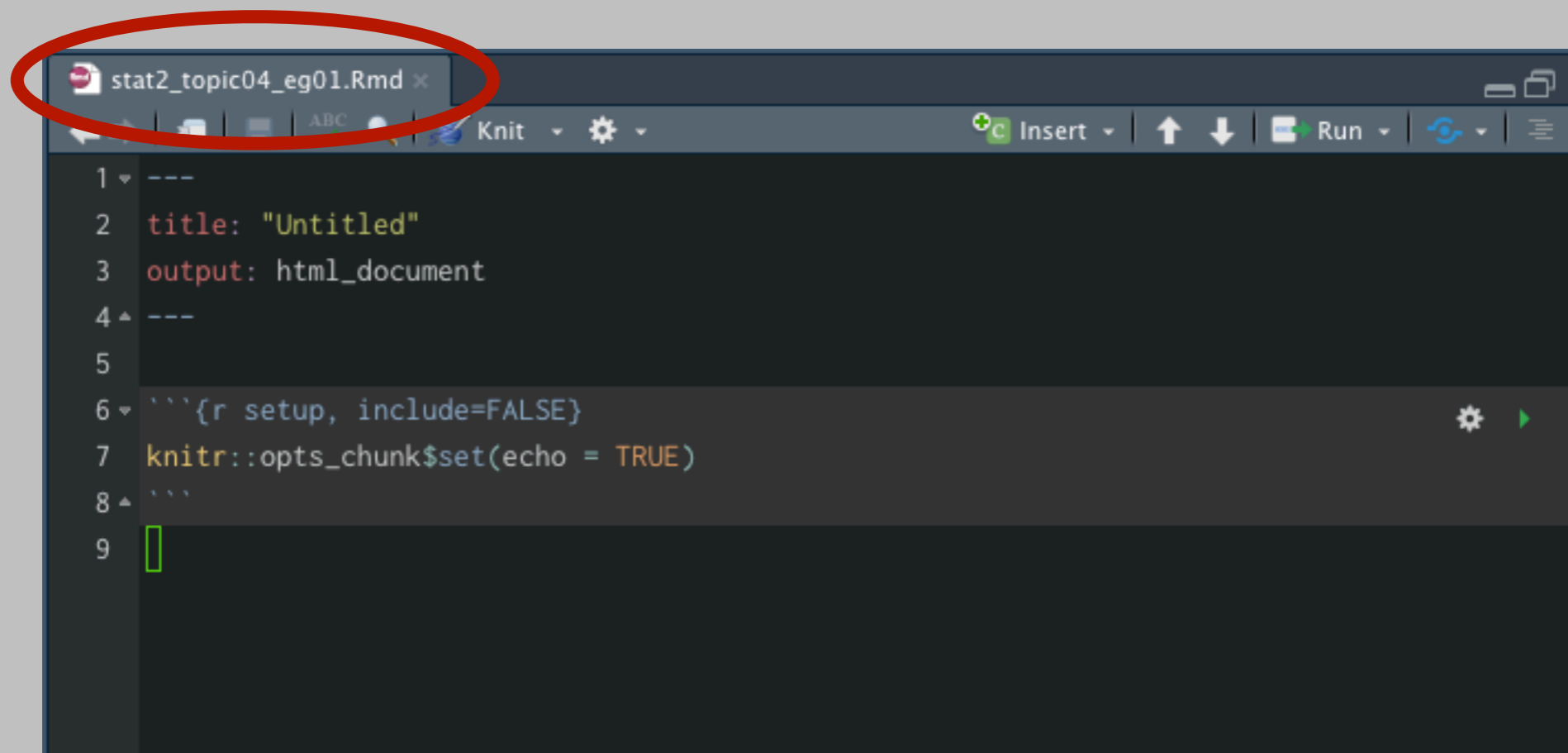
- 下の図のようなファイルが表示される
- 冒頭部分はサンプルコードからコピペする



```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8 ```
9
10 ## R Markdown
11
12 This is an R Markdown document. Markdown is a simple formatting syntax for authoring
13 HTML, PDF, and MS Word documents. For more details on using R Markdown see
14 <http://rmarkdown.rstudio.com>.
15
16 When you click the Knit button a document will be generated that includes both
17 content as well as the output of any embedded R code chunks within the document. You
18 can embed an R code chunk like this:
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
```


R Markdown ファイルの保存 (1)

- [Ctrl/Cmd + S] でファイルを保存する
 - ▶ ファイル名を付けるように促されるので、**ファイル名拡張子を忘れずに名前を付ける**
 - ファイル名の例：stat2_topic04_eg01.Rmd



```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8 ```
9
```

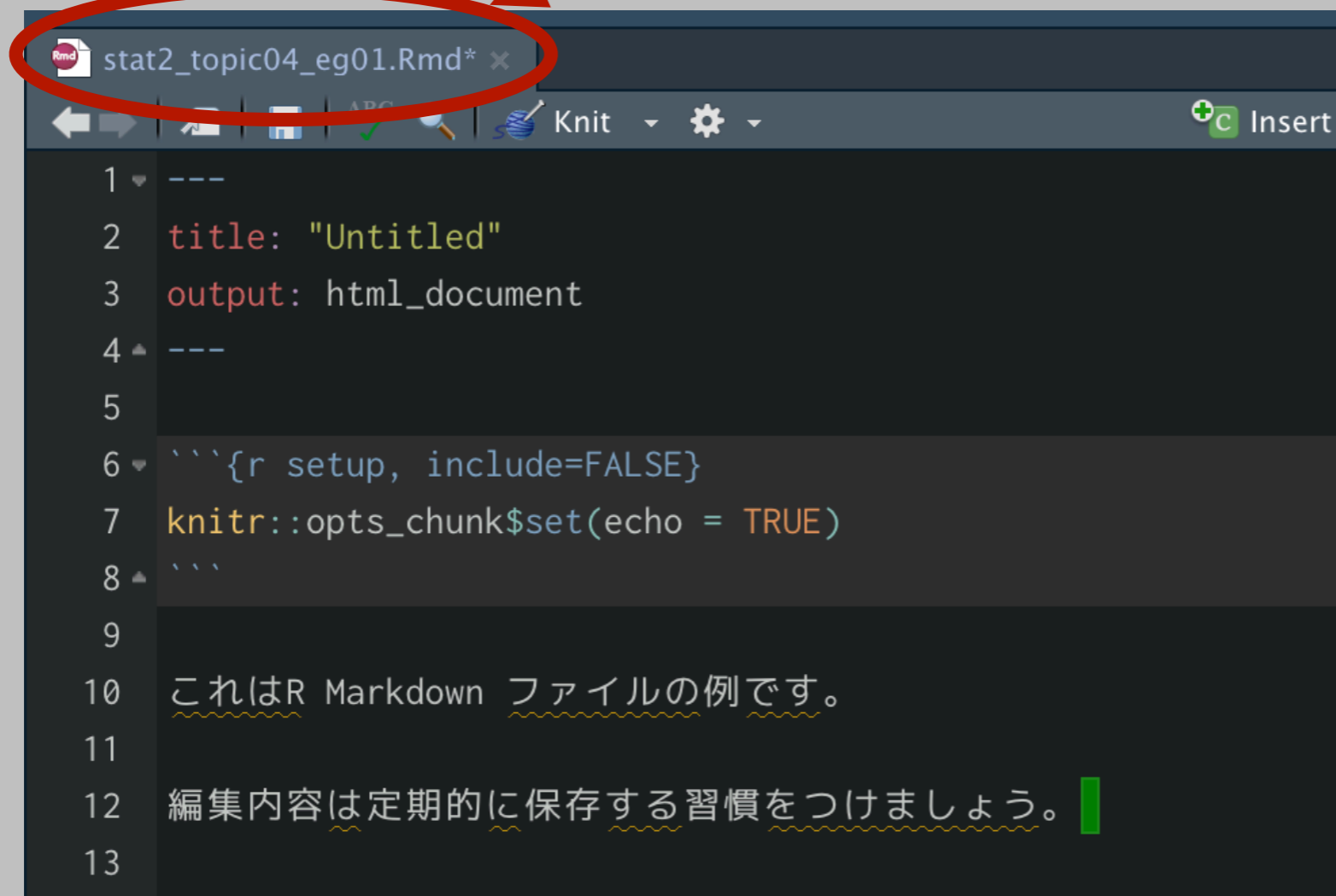
ファイル名の付け方（復習）

- ファイル名は以下のルールで付ける
 - ▶ 半角の英数字とアンスコ「_」、ハイフン「-」のみ
 - **全角文字は絶対に使わない**（日本語はダメ！）
 - **ファイル名にスペースを使わない**
 - ▶ 最初の1文字はアルファベット
 - ▶ ファイルの中身がわかる名前を付ける
- 提出物に関しては、さらに次のルールを守る
 - ▶ 指示がある場合はそれに従う
 - ▶ 指示がなくても、受け取る人がファイルを区別できる名前を付ける

R Markdown ファイルの保存 (2)

- ファイルを編集する際は、定期的に [Ctrl/Cmd + S] を押してファイルを保存する

保存されていない編集内容があると、
ファイル名が青文字になり、* が付く

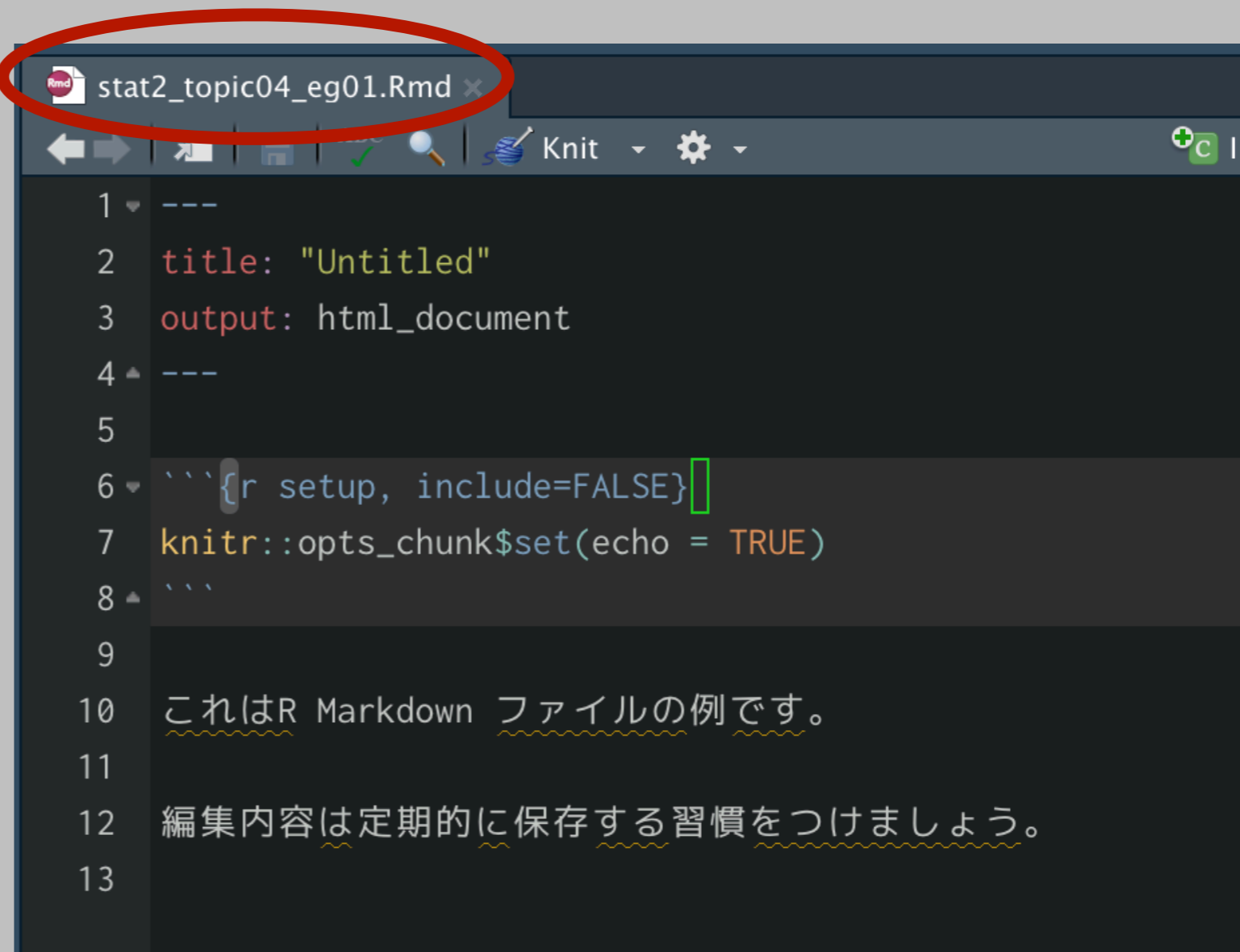


```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8 ```
9
10 これはR Markdown ファイルの例です。
11
12 編集内容は定期的に保存する習慣をつけましょう。
13
```

R Markdown ファイルの保存 (2)

- ファイルを編集する際は、定期的に [Ctrl/Cmd + S] を押してファイルを保存する

[Ctrl/Cmd + S] で保存すると、* が消え、ファイル名が白文字に



```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8 ```
9
10 これはR Markdown ファイルの例です。
11
12 編集内容は定期的に保存する習慣をつけましょう。
13
```

R Markdown ファイルの 書き方

1. YAML ヘッダ

YAML ヘッダ

- YAMLヘッダ：R Markdown ファイルの冒頭にある3つの連続するハイフン「---」のペアで囲まれた部分
- ファイル自体の情報や出力の方法（HTMLとPDFのどちらを作るか）をコントロールする
- YAML (YAML Ain't Markup Language)：データ形式の一種
 - ▶ 授業では必要最小限のみ扱う
 - ▶ 詳しくは、<https://magazine.rubyist.net/articles/0009/0009-YAML.html> を参照

YAMLヘッダの編集

- 基本的には、配布したサンプルの真似（コピペ）をすれば良い
- ただし、以下の要素は**毎回必ず自分で書き換えること**
 - ▶ title: "文章のタイトル"（ファイル名ではない！）
 - ▶ subtitle: "文章のサブタイトル"（必要な場合のみ）
 - ▶ author: "自分の氏名"
 - この授業の課題を出す場合は氏名と学籍番号で、「矢内 勇生 (1234567)」のようにするのが最善（次のページの例を参照）

YAMLヘッダの例

- 右のように書く
 - ▶ 1行目と27行目に YAML
ヘッダの区切りがある
- インデント（字下げ）によってブロックを作っていることに注意
 - ▶ ブロックごとのインデントは必須！（正しくインデントしないと動かない）

```
stat2_topic04_eg01.Rmd x
← → | 📄 | 📁 | ABC | 🔍 | 🌐 Knit | ⚙️
1 ---
2 title: "R Markdown による文章作成の例"
3 subtitle: "How Fantastic R Markdown Is!"
4 author: "矢内 勇生 (22222222)"
5 date: "`r Sys.Date()`"
6 output:
7   pdf_document:
8     highlight: tango
9     latex_engine: xelatex
10    number_sections: true
11    toc: false
12  html_document:
13    highlight: tango
14    theme: united
15    toc: true
16    smart: false
17  header-includes:
18    - \usepackage[indentfirst]
19    - \parindent = 1em
20    - \usepackage{dcolumn}
21    - \newcolumntype{.}{D{.}{.}{-1}}
22    - \usepackage{caption}
23    - \captionsetup[table]{name=表}
24    - \captionsetup[figure]{name=図}
25  documentclass: bxjsarticle
26  classoption: xelatex, ja=standard
27 ---
28
```


YAMLヘッダの書き方の注意

- 各要素名の後にコロン「:」が必要
 - ▶ コロンの後に**半角スペース**が1つ以上必要（ブロックを構成するときは、改行だけでOK）
- 要素の内容に日本語を使う場合は、半角の引用符で囲んでおくのが安全

半角スペース2つの例

(全角スペースは絶対ダメ！)

```
title:  "Rはともだち"  
author: "矢内 勇生"
```

半角スペース1つの例

半角二重引用符

R Markdown ファイルの 書き方

2. 文章

文章

- YAML ヘッダより後の行に、文章をそのまま書く
- 日本語も使える
 - ▶ ただし、RStudio Cloudで日本語を使うと、後でknitするときに失敗する（現在のところ解決策なし）
 - ▶ 大学の（自分の）パソコンにインストールされたRStudio を使ったほうが良い
 - ▶ この授業ではすべて英語でレポートを書いてもOK
- Markdown 記法が使える
- コメントではないので、すべての文に # を付けない！

よく使うマークダウン記法 (1)

- ハッシュ記号「#」で文章の構造を決める
 - ▶ Rスクリプトにおけるコメントとは役割が異なるので
注意
- #: 文章の中の最大の構成単位 (HTML の `<h1>`)
 - ▶ レポートでは「節 (section)」
- ## : その次の構成単位 (`<h2>`, 小節 [subsection])
- ### : その次の構成単位 (`<h3>`, 見出し [subsubsection])

よく使うマークダウン記法 (2)

- 番号なしの箇条書きは、アスタリスク「*」またはハイフン「-」で作る
- 番号付きの箇条書きは、「1.」で作る
 - ▶ 番号は、1, 2, … と増やしてもいいし、全部1でも良い
(自動で正しい番号が振られる)
 - 全部1にしておくと、後で並べ替えるのが楽
- 箇条書きの階層はインデントによるブロック化で作る
- それぞれ、**記号の直後に半角スペースが必要**

よく使うマークダウン記法 (3)

- 文字をイタリック (斜体) にするときは、「*」または「-」で挟む (例: *This sentence will be Italic.*)
 - ▶ 日本語はイタリックにしない!
- 太字にするには、「**」または「--」で挟む (例: **この文は太字になる**)
- イタリックかつ太字にするには、「***」または「---」で挟む
 - ▶ 日本語では使わない
- それぞれ、記号の直前・直後にスペースを入れないように注意

よく使うマークダウン記法 (4)

- リンクの貼り方：[表示する文](URL)
 - ▶ 例：[矢内のwebsite](https://yukiyanai.github.io)
- 画像の貼り方：![コメント](画像ファイルのアドレス)
 - ▶ 例：![三毛猫](pics/calico.png){width=50%}

文章内にRコードを埋め込む

- Rコードをバックティックマーク「```」で囲んで最初に「`r`」と書く

- 例： `1+2`の計算結果は ``r 1 + 2`` です。

- ▶ 出力： `1+2`の計算結果は3です。

- 例： 円周率は約 ``r round(pi, digits = 2)`` です。

- ▶ 出力： 円周率は約3.14です。

数式：LaTeX 記法が使える*

- 文章内の数式：「\$」で囲む（\ は日本語環境では¥）

▶ 例： `x_i`の合計は`$$\sum_{i=1}^n x_i$` です。

▶ 出力： x_i の合計は $\sum_{i=1}^n x_i$ です。

- 数式ブロック：「\$\$」で囲む

▶ 例： `x_i`の合計は
`$$\sum_{i=1}^n x_i$$`
です。

▶ 出力： x_i の合計は
$$\sum_{i=1}^n x_i$$

です。

LaTeX の参考書



R Markdown ファイルの 書き方

3. コードチャンク

コードチャンク (code chunk)

- バックティックを3つまとめた「````」のペアで囲んだ部分を「コードチャンク」と呼ぶ
- チャンク冒頭に言語の指定が必要
 - ▶ Rの場合は `{r}` と書く
 - ▶ R以外の言語も書ける！（今のところ他に使えるのはPythonのみ）

これは通常の文章である。コードチャンクは、

```
`` {r}
```

```
x <- 1:10
```

```
mean(x)
```

```
``
```

のように書く。

RStudioでのコードチャンクの表示

```
37
38 これは通常の文章である。コードチャンクは、
39 ```{r}
40 x <- 1:10
41 mean(x)
42 ```
43 \noindent
44 のように書く。
45
```

- 上の例の39-42行がコードチャンク
 - ▶ 40-41行にRコードを書く
- コードチャンクの背景色は文章の部分と少し異なる
- チャンクの右上にボタンが表示される

コードチャンクの実行結果

```
37
38 これは通常の文章である。コードチャンクは、
39 ```{r}
40 x <- 1:10
41 mean(x)
42 ```
43 \noindent
44 のように書く。
45
```

- この部分の出力 (PDF) :

```
これは通常の文章である。コードチャンクは、

x <- 1:10
mean(x)

## [1] 5.5

のように書く。
```

注：この例では、\noindent で字下げしない（新しい段落を作らない）ようにしている

コードチャンクの作り方

- 「`」を1つずつ入力することもできるが、**この方法は避けるべき**
 - ▶ 間違えやすい
 - ▶ **面倒くさい**
- RStudio の**ショートカット** : 3つのキーを同時に押す
 - ▶ macOS: `command + option + I`
 - ▶ Windows: `Ctrl + Alt + I`

コードチャンクの実行方法

- 基本はRスクリプトでのコードの実行方法と同じ
 - ▶ 実行する行にカーソルをおいた状態で `cmd + return` (macOS) または `Ctrl + Enter` (Windows)
- チャンク内のコード全体を一挙に実行したい場合は、
 - ▶ macOS: `command + shift + return`
 - ▶ Windows: `Ctrl + Shift + Enter`
 - ▶ または、チャンク右上の実行ボタン (三角マーク)

チャンクオプション

- チャンクごとにオプションを設定できる
 - ▶ すべてのチャンクに一律に適用するオプションは、グローバルオプションとして指定できる
 - ▶ グローバルオプションがあっても、チャンクごとのオプションが優先される
- オプションで指定できることの例
 - ▶ コードを実行しない（見せるだけ）
 - ▶ コードを見せない（レポートには結果のみ表示）
 - ▶ メッセージの表示・非表示
 - ▶ 図の大きさ
 - ▶ 図のキャプション

- 詳しくは、<https://teramonagi.hatenablog.com/entry/20130615/1371303616> を参照

R Markdown ファイルを
knit する

レポートを出力する

- R Markdown を 「“knit” する (編む) 」 ことで、Rの (コードと) 実行結果を埋め込んだ文書が作れる
- 主な出力
 - ▶ HTML
 - ▶ PDF
 - ▶ (MS Word)
- rmarkdown パッケージを使う
- RStudio に knit ボタン (毛糸と編み針のマーク) がある！

正しい出力のために

- R Markdown ファイルを「正しく」書く必要がある
 - ▶ よくあるエラー
 - YAMLヘッダがない
 - YAMLヘッダのスペルミス
 - 閉じていないコードチャンクがある
- Rコードも「正しい」必要がある
 - ▶ コードにエラーがあると knit できない (チャンクオプションでエラーも出力できるようにした場合を除く)
- YAMLヘッダで出力をコントロールする

HTMLファイルを作る

- YAMLヘッダに次の1行を書く
　　<pre>output: html_document</pre>
- オプションを指定するとき
　　は、右の例のように output:
　　のブロックを作る

```
---
title: "HTML出力の例"
author: "矢内 勇生"
date: "`r Sys.Date()`"
output:
  html_document:
    theme: united
    highlight: tango
    toc: true
    smart: false
---
```

PDFファイルを作る

- YAMLヘッダを右のように書く
 - ▶ HTMLとは異なり、1行の命令だけでは動かない
- LaTeX の設定状況によって色々
とカスタマイズが必要なので、
困ったら質問を
- グローバルチャンクオプション
で `dev = "cairo_pdf"` を指
定する必要あり (ウェブ資料を
参照)

```
---  
title: "PDF出力の例"  
author: "矢内 勇生"  
date: "`r Sys.Date()`"  
output:  
  pdf_document:  
    latex_engine: lualatex  
    toc: false  
documentclass: bxjsarticle  
classoption: lualatex  
---
```

複数の出力形式

- YAMLヘッダに複数の出力オプション (output) を共存させることが可能
- Knit ボタンを押すと、上に書いてあるオプションが利用される
 - ▶ 右の例では HTML が優先される
 - ▶ Knitボタンの右端にあ三角ボタンを押せば、特定の出力が選択できる
 - 右の例の状態では PDF に knitすると、YAML ヘッダのoutputの順番が入れ替わる (自分でやってみよう)

```
---  
title: "出力指定の例"  
author: "矢内 勇生"  
date: "`r Sys.Date()`"  
output:  
  html_document:  
    theme: united  
    highlight: tango  
    toc: true  
    smart: false  
  pdf_document:  
    latex_engine: lualatx  
    toc: false  
documentclass: bxjsarticle  
classoption: lualatex  
---
```

PDF ファイルに knit できない?

- 授業の課題を提出するためのとりあえずの解決策
 - ▶ R Markdown ファイルを knit してHTMLファイルを作る
 - ▶ HTMLファイルを web ブラウザ (Chrome, Firefox, or Safari) で開く
 - ▶ メニューから [ファイル] -> [プリント (印刷)] と進む
 - ▶ プリントのメニューで [PDF] -> [PDF として保存] を選ぶ
 - ▶ 適切なファイル名をつけて保存する (拡張子は .pdf)

R Markdown Cookbook

- 英語が読めるなら、rmarkdown の作者である Yihui Xie 氏らが書いた本がオンラインで無料で読める
 - <https://bookdown.org/yihui/rmarkdown-cookbook/>
- 片桐智志氏による日本語訳プロジェクトも進行中
 - <https://gedevan-aleksizde.github.io/rmarkdown-cookbook/>

課題

- 以下のウェブページにある提出課題1を実行し、KUTLMS に提出しなさい
 - ▶ https://yukiyanai.github.io/jp/classes/stat2/contents/R/stat2_2021_hw01.html
- **提出期限：2021年4月28日（水）正午**（日本時間）
 - ▶ 当然だが、締切より前であればいつ出しても良い：早めに取り組もう！
- **指示をよく読むように！**

次回予告

5. ggplot2 入門