



# 統計学2

## 2. Rの基本操作

やない ゆうき  
矢内 勇生



<https://yukiyanai.github.io>



[yanai.yuki@kochi-tech.ac.jp](mailto:yanai.yuki@kochi-tech.ac.jp)



# このトピックの目標

- RとRStudio の基本的な使い方を覚える
  - ▶ Rとは何かを知る
  - ▶ RStudio とは何かを知る
  - ▶ RStudio を使ってRを操作する基本的な方法を身につける

Rとは何か



# とは？

統計解析のためのプログラミング言語

- オープンソース
  - ▶ 無料
  - ▶ プログラムの中身が読める
  - ▶ 柔軟
  - ▶ 世界中のユーザがR用のパッケージを開発
  - 統計学のスタンダード
  - ▶ 知識の共有を促進
  - ▶ 最新の手法が実装される
  - 綺麗な図表が作れる
- ★とにかく便利！

★使わない理由が見当たらない！

# R Studio<sup>®</sup> とは？

- Rのための統合開発環境 (IDE)
  - ▶ 無料！
  - ▶ Rの使用が楽になる：特に初心者にオススメ！
    - コードの記録：Rコードを記録しながら実行できる
    - コードの補完：途中まで入力して <Tab> で補完
    - プロジェクト機能：ファイルの管理が容易

# RとRStudioの導入

# CRAN

- CRAN (Comprehensive R Archive Network)
  - ▶ R本体とRに追加するパッケージの置き場所
  - ▶ 世界中にミラーサイトがあるので、自分に近いところを利用する
  - 統計数理研究所のCRAN ミラー：  
<https://cran.istm.ac.jp/>
  - ▶ 自分で作ったパッケージ CRANに登録することもできる！

# RStudio Cloud: インストールいらずのRStudio

- RStudio Cloud (<https://rstudio.cloud/>) : webブラウザで RStudio が使える
  - ▶ **インストール不要** : アカウントを作ればファイルも保存できる
  - ▶ 自分のプロジェクトを公開できる
- 注意点：
  - ▶ 安定したインターネット接続が必要
  - ▶ 日本語を使うのが困難 (不可能ではない)
  - ▶ 2020年7月に一部を除いて有料化
- この授業では使用を (禁止はしないが) **推奨しない**

# 詳しくは、以下の資料を参照

<https://yukiyanai.github.io/jp/resources/>

- OSによってインストール方法が異なるので注意
  - ▶ Linux用
  - ▶ macOS用
  - ▶ Windows用

# インストールする前に確認すること

- ・パソコン自体のユーザ名に全角文字（特に全角スペース）がないか？
  - ▶ダメなユーザ名：「矢内勇生」
  - ▶最悪なユーザ名：「矢内 勇生」（姓と名の間に全角スペース）
  - ▶良い例：「yuki」
- ・**ユーザ名に全角文字があると、Rをまともに使えない！**
  - ▶Rに限らず、他のプログラミング言語を使う際にもつまづくポイントに
- ・解決策：パソコンを初期化するしかない…
  - ▶とりあえずの回避策は、インストール解説資料に掲載

# インストール：macOS

## 1. Rのインストール

- ▶ [https://cran.rsm.ac.jp/\\_](https://cran.rsm.ac.jp/) で、 “Download R for (Mac) OS X” -> “R-4.0.2.pkg” （数字は異なる可能性あり） を順にクリック
- ▶ ダウンロードしたファイルをクリックして指示通りにインストール（ただし、GUI はいらないので、途中の [カスタマイズ] で GUI のチェックを外した方がよい）

## 2. Xcodeのインストールと起動

- ▶ AppStore から Xcode を入手（無料）して、**一度再起動する**

## 3. XQuartz のインストール（必須ではない）

- ▶ [https://www.xquartz.org/\\_](https://www.xquartz.org/) でXQuartz-2.7.11.dmg をクリックしてダウンロードし、インストール

## 4. RStudio のインストール

- ▶ <https://rstudio.com/products/rstudio/download/#download> で、 ”Download RStudio Desktop” の下にあるボタンに “FOR MAC” と書かれていることを確認してクリックし、ダウンロードしてインストールする

# インストール：Windows

## 1. Rのインストール

- ▶ <https://cran.rsm.ac.jp/> で、 “Download R for Windows” -> “install R for the first time” -> “Download R 4.0.2 for Windows”（数字は異なる可能性あり）を順にクリック
- ▶ ダウンロードしたファイルをクリックして指示通りにインストール
  - 途中でパス (path) を設定するかどうか聞かれるので、必ず「はい」を選択

## 2. Rtools のインストール

- ▶ <https://cran.r-project.org/bin/windows/Rtools/> で、Rtools40.exe（上でインストールしたRのバージョンに合ったもの）をクリックしてダウンロードし、インストールする

## 3. RStudio のインストール

- ▶ <https://rstudio.com/products/rstudio/download/#download> で、 "Download RStudio Desktop" の下にあるボタンに “FOR WINDOWS” と書かれていることを確認してクリックし、ダウンロードしてインストールする

# インストール：Linux

- Ubuntuの場合

1. Rのインストール：ターミナルで以下を実行

```
sudo apt-get install -y r-base-core
```

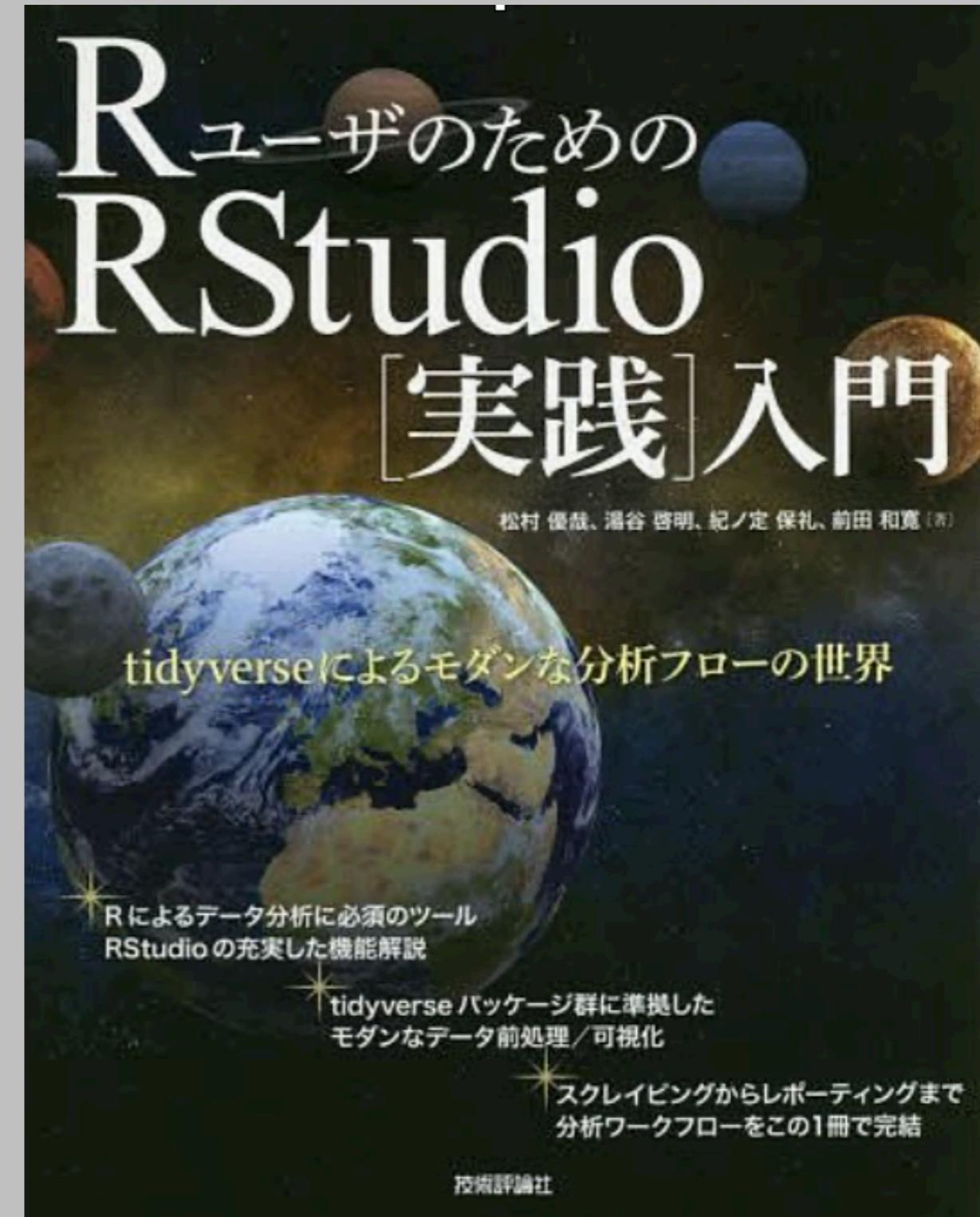
- 最新版が必要な場合は、/etc/apt/sources.list に CRAN のミラーサイトを追加する必要あり

2. RStudioのインストール：.deb ファイルをRStudio からダウンロードし、インストール

- その他のLinux

▶ 自力で頑張って！

# R導入の手引き書

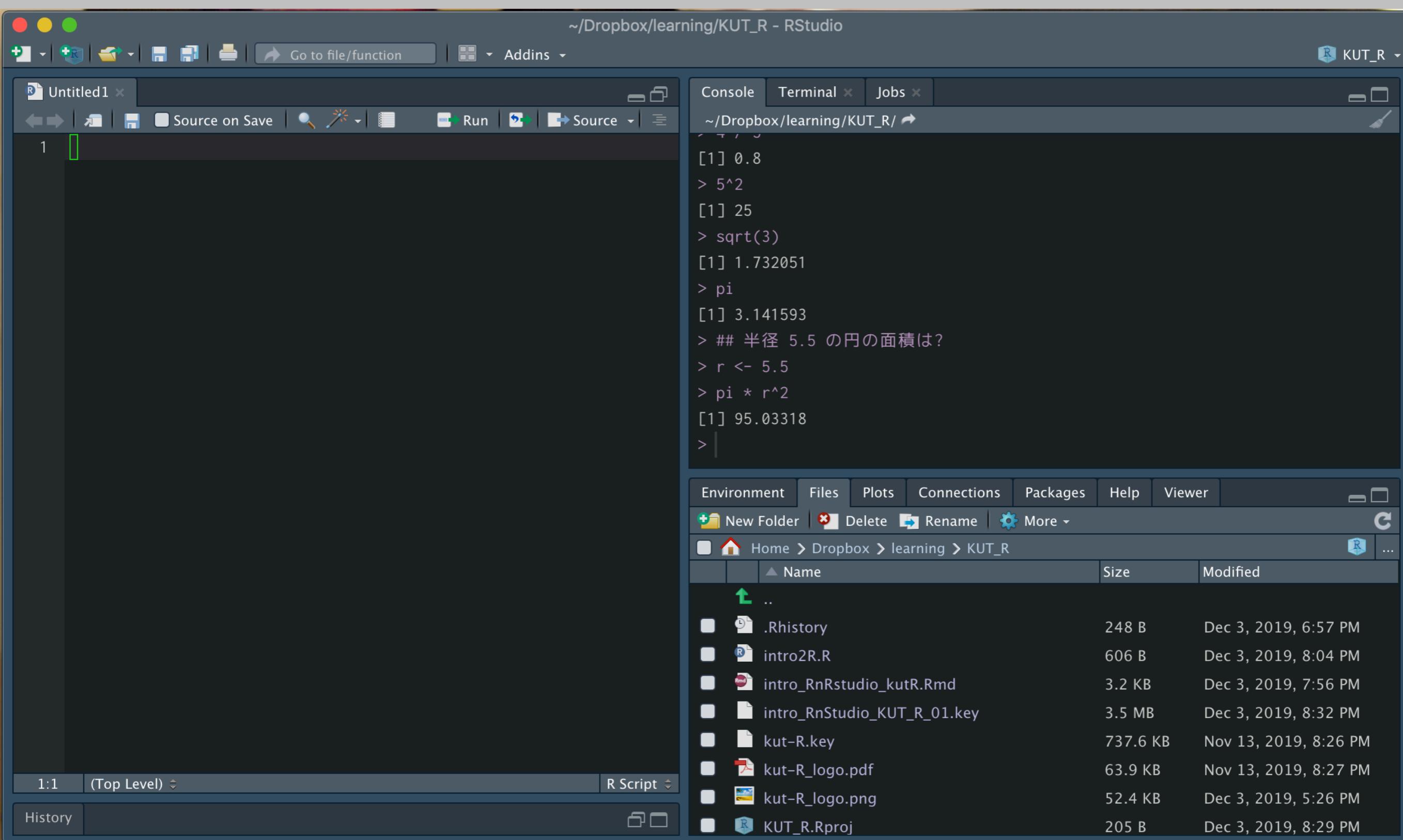


R と RStudio を  
使ってみよう

# Rの使い方

- RStudio を起動する
  - ▶ R自身を起動する必要はない：RStudio内でRが動く
- コードを**すべて保存**して実行する：**再生性**の確保（トピック4で説明する）
  - ▶ Rスクリプト（拡張子が.R のファイル）に保存
    - Rコードを書くためのファイル
  - ▶ Rマークダウン（拡張子が.Rmd のファイル）に保存
    - こちらの方がオススメ！（トピック4で説明する）
    - Rのコードと文章の両方を自由に書ける
    - これだけで論文 자체が書ける！

# RStudioのスクリーンショット



# RStudioの基本

- RStudioの左上のウィンドウで、Rスクリプト (file\_name.R)  
または R Markdownファイル (file\_name.Rmd) を開く
- Rスクリプト (またはRmd) にRの命令を書いて**保存**する
- Rスクリプト (またはRmd) ファイル内で、実行したい行（ブロック）にカーソルを合わせて、以下のショートカットで実行
  - ▶ macOS: command + return
  - ▶ Windows: Ctrl + Enter
- Console で実行結果を確認する (Rmd ファイルの場合は、ファイルの中に埋め込み表示することもできる)

# ファイル名の付け方

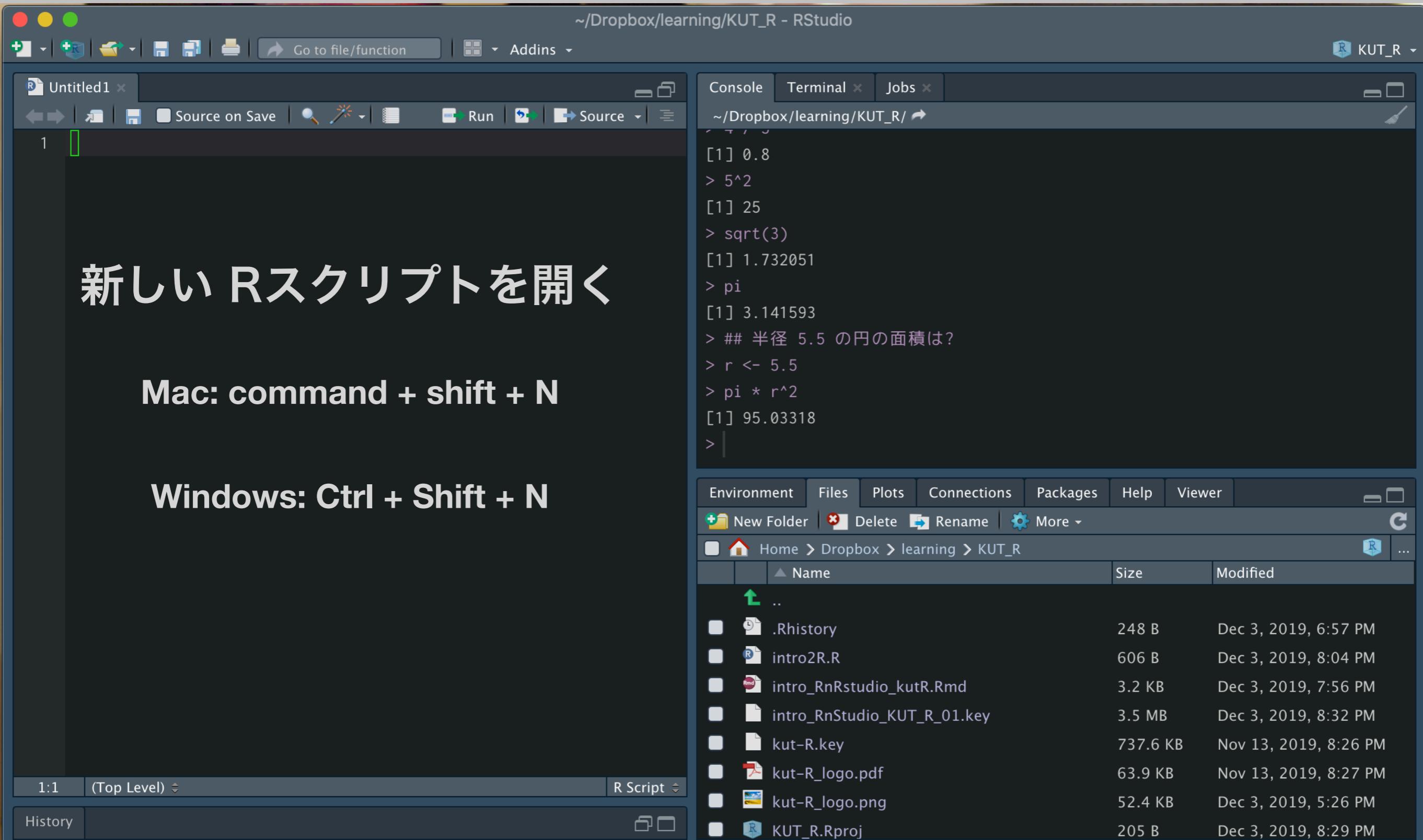
- ・ ファイル名は以下のルールで付ける
  - ▶ 半角の英数字とアンスコ「\_」、ハイフン「-」のみ
    - 全角文字は絶対に使わない（日本語はダメ！）
    - ファイル名にスペースを使わない
  - ▶ 最初の1文字はアルファベット
  - ▶ ファイルの中身がわかる名前を付ける
- ・ 提出物に関しては、さらに次のルールを守る
  - ▶ 指示がある場合はそれに従う
  - ▶ 指示がなくても、受け取る人がファイルを区別できる名前を付ける

# RStudioの基本

新しい Rスクリプトを開く

Mac: command + shift + N

Windows: Ctrl + Shift + N



# RStudioの基本

The screenshot shows the RStudio interface with the following components:

- Script Editor (Left Panel):** Displays an R script named "intro2R.R". The code includes basic operations like addition and multiplication, and calculates the area of a circle with radius 5.5. A red overlay message at the bottom of the editor says: "# 開いたらすぐに Ctrl + S で名前を付けて保存する (最後の .R を消さないように！)".
- Console (Top Right Panel):** Shows the output of the R code run in the console, including calculations like  $5^2$ ,  $\sqrt{3}$ , and  $\pi$ .
- File Browser (Bottom Right Panel):** Shows the directory structure: Home > Dropbox > learning > KUT\_R. It lists files like ".Rhistory", "intro2R.R", and "KUT\_R.Rproj".
- Text Overlay (Center):** Large text in the center of the screen reads "Rスクリプト (拡張子が .R のファイル)" and "コンソール Rを実行する場所".

# RStudioの基本

The screenshot shows the RStudio interface. The top bar displays the path `~/Dropbox/learning/KUT_R - RStudio`. The left pane is the `Code Editor` containing an R script with the following code:

```
8 # R の基本操作
9 ## R を電卓として使う
10
11 1 + 1
12 3 * 4
13 4 / 5
14 5 ^ 2
15 sqrt(3)
16 pi
17
18 ## 半径 5.5 の円の面積は?
19 r <- 5.5
20 pi * r^2
21
22 ## 数列を作る
23 1:10
24 1990:2020
25
26
27
28
29 ## データフレーム
30 df <- data.frame(x = 1:3,
31                   y = 11:13)
```

The right pane is the `Console`, showing the execution results:

```
[1] 0.8
> 5^2
[1] 25
> sqrt(3)
[1] 1.732051
> pi
[1] 3.141593
> ## 半径 5.5 の円の面積は?
> r <- 5.5
> pi * r^2
[1] 95.03318
>
```

Annotations in Japanese are overlaid on the interface:

- A large white arrow points from the text "Rの命令をここに書いて保存 (再現が可能!)" in the bottom-left towards the `Code Editor`.
- A large white arrow points from the text "実行したい行にカーソルを合わせて Ctrl + Enter で実行" in the bottom-center towards the `Code Editor`.
- A large white arrow points from the text "実行結果を表示" in the bottom-right towards the `Console`.

# Rスクリプトの基本ルール

- ・拡張子は .R (ファイル名の例: `prelim_eda.R`)
  - ▶ 「拡張子を常に表示」するようパソコンの設定を！
- ・1行に1つの命令
- ・# (ハッシュ記号) の後にコメント : Rは # 以降を無視 (改行で元に戻る)
- ・コードが読みやすいように適宜スペースを使う : **スペースは必ず半角**に
- ・1つの命令が複数行にわたるとき
  - ▶ 2行目以降をインデント (字下げ) : 命令が続いていることが見た目でわかる
  - ▶ 入れ子構造 (複数の階層) の命令は、階層ごとにインデント幅を変える : 深い階層ほど右にずらす
  - ▶ RStudio は勝手にインデントしてくれる (インデント幅は設定可能)

# ショートカットキーを使おう

- Rに限らず、パソコンを使いこなすにはショートカットキーを覚えるのが大切
- ショートカットキーを覚えると
  - ▶ 入力間違が減る
  - ▶ 入力が速くなる
  - ▶ キーボードから手を離す必要がない
    - 両手がホームポジションから離れないでさらに間違が減る
- ★タッチタイプができない人は、今のうちに覚えよう！

ERGOTRON

Type-S

HHKB  
Professional 2



# ショートカットの表記

- OSによってややや異なるので注意
  - ▶ mac でよく使うキー : command (Cmd と表記することも)
  - ▶ Windows でよく使うキー : Ctrl (コントロール)
- 「Ctrl + Enter」の意味
  - ▶ Ctrl と Enter を同時に押す
    - 同時というのは、Ctrl を押したまま Enter を押すということ
- Cmd/Ctrl と書いてあったら、macOSではCmd を、Windows では Ctrl を押すという意味（自分にあったほうを選ぶように）

# RStudio でよく使うショートカット

- ファイル関係
  - ▶ ファイルの保存 : Cmd/Ctrl + S
  - ▶ 新しい Rスクリプトの作成 : Cmd/Ctrl + Shift + N
- Rのコード
  - ▶ カーソル行の実行 : Cmd/Ctrl + Return/Enter
  - ▶ コードチャunkの実行 : Cmd/Ctrl + Shift + Return/Enter
  - ▶ <- の入力 : option/Alt + - [マイナス]
  - ▶ %>% の入力 : Cmd/Ctrl + Shift + M

# デモンストレーション

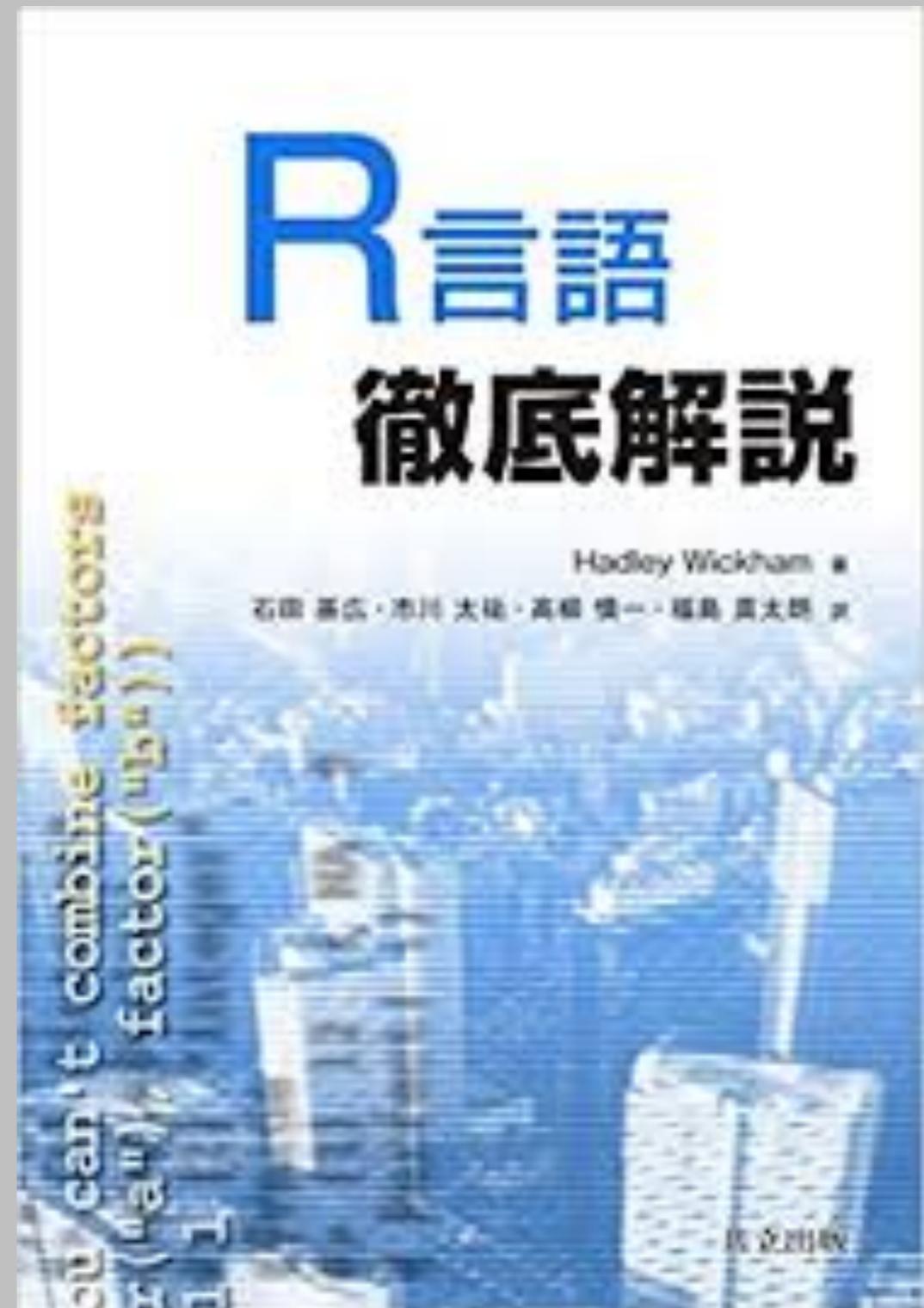
参考 :

- <http://yukiyanai.github.io/jp/classes/stat2/contents/R/introduction-to-R.html>
- <https://rstudio.cloud/project/762403>

 と  Studio<sup>®</sup> と

仲良くなろう！

# Rは言語



# 言語習得のコツ

- ・目的を明確にする
- ・とにかく使ってみる！
  - ▶ 毎日使う！
  - ▶ 間違ってもやってみる！
  - ▶ 使える人と付き合う！

# プログラミングの注意

- 最初は少しづつ進める
  - ▶ 最初から長いコードは書けない
- 試行錯誤が重要
  - ▶ サンプルコードを少しだけ変えてみる
  - ▶ 変更したコードを実行する前に結果を予測する
  - ▶ コードを実行し、予測と実行結果を比較する
- エラー (Error!) はともだち！
  - ▶ エラーメッセージを「読む」ことが大事
  - ▶ エラーで指摘されたところを直せば良い！

# Rで何がしたいのか？

- Rは道具に過ぎない

▶ Rを使って何をする？

- やりたいことがなければ、Rを使う意味はない
- Rで何ができるかがわからないと、Rで何をするか決められない

★ Rの使い方と分析法（統計学、計量分析）と**実質科学的知識**（経済学、経営学、心理学、政治学、etc.）をバランスよく勉強することが必要！

# RStudioを使う！

- データ分析に関わること
  - ▶ データの収集
  - ▶ データの前処理
  - ▶ データ分析
  - ▶ データ可視化（図の作成）
  - ▶ 分析結果のまとめ（表の作成）
  - ▶ 論文執筆
- データ分析以外
  - ▶ ターミナルが使える
  - ▶ YouTubeが見られる
  - ▶ メールが送れる
  - ▶ tweet できる
  - ▶ etc.

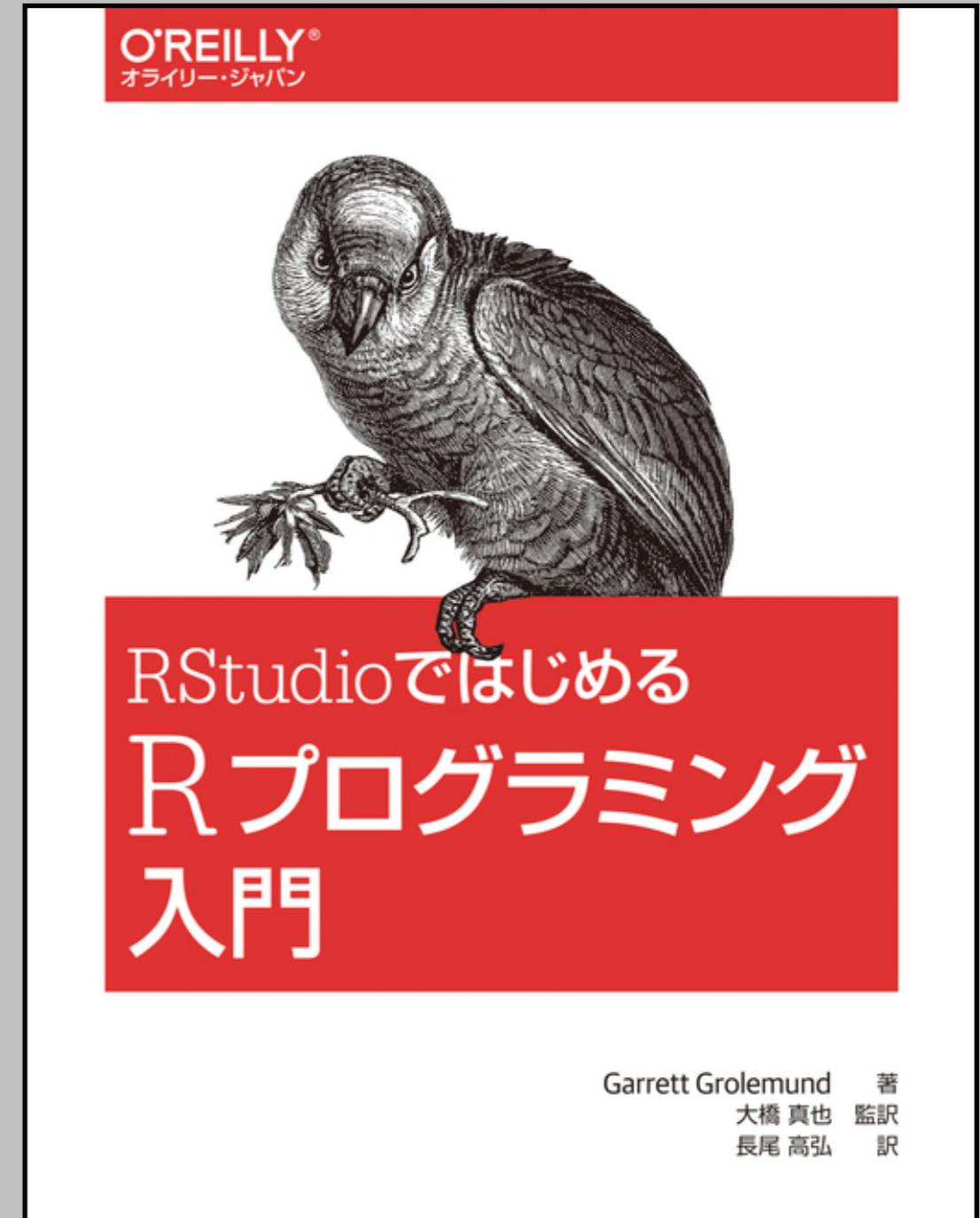
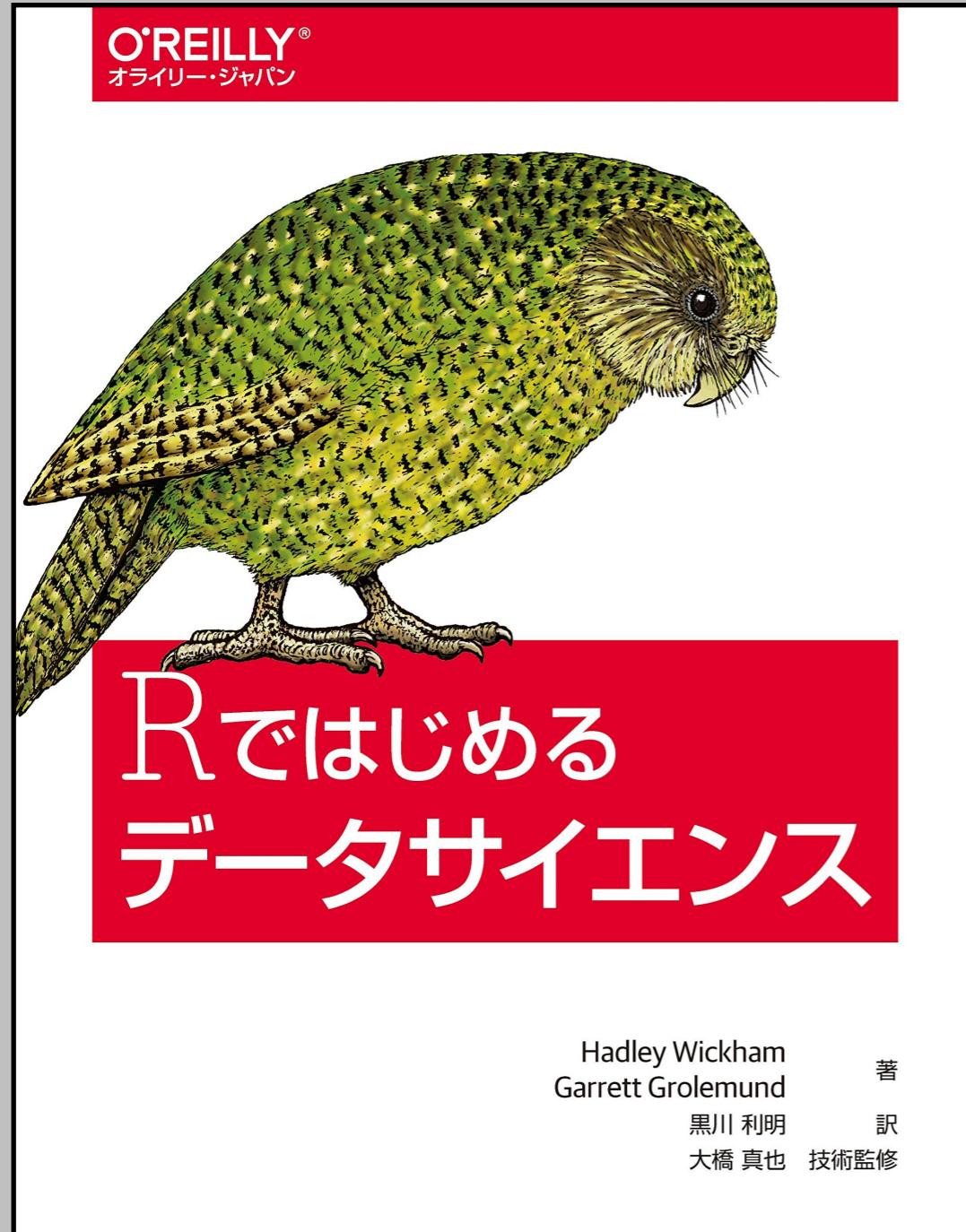
# Rが使える人と付き合う

- 勉強会に参加しよう！
  - ▶ KUT.R
  - ▶ 勉強グループを作る
- 詳しい人に訊いてみよう！
  - ▶ 担当教員に質問する
  - ▶ r-wakalang: <https://github.com/tokyor/r-wakalang>
  - ▶ stack overflow: <https://ja.stackoverflow.com/questions/tagged/r>

# Rについて質問する場合の注意

- ・以下のような質問には対処不可能（原因を特定できないので、助けたくても助けられない）
    - ▶ 「指示通りにやったのに動かない」
    - ▶ 「エラーが出ました」
  - ・実際に自分で入力したコード「全体」とコンソールに表示された出力「全体」のすべてをセットでSlackにアップするように（添付ファイルをメールで送られると迷惑メールだと思って読まない）
- ★同じような質問が複数の受講生から出た（出ることが想定される）場合には、個別に回答せず、クラス全体に向けて回答する

# 参考書



# コーディングスタイル

# コーディングスタイル

- Rコードの書き方については、教科書、副読本、ウェブ資料のコードを真似すれば、読み易いコードが書ける
  - ▶ 世の中には「悪い見本」もあるので注意
  - ▶ 担当教員のコーディングスタイルが最善だと言うつもりはないが、それなりに良いはず
    - 「もっとこうしたほうがいいのでは」という提案や、「なぜそうするのか知りたい」という質問はいつでも歓迎

# コードを書くときに考慮すべきこと

- ・可読性：適切なスペーシング、改行、字下げ（ブロック化）
- ・オブジェクト名の一貫性
  - ▶ スネークケース：linear\_model
  - ▶ キャamelケース：linearModel
- ・来週、来月、来年、5年後、10年後、…に読んで理解できるコードを書く：**コメントをたくさん書く！**
- ・コメント（Rmd の場合は普通の文章）が少なすぎない？  
(決して多すぎることはない！)

# 人間にやさしいコードを！

- ・「正しい」コードは優れたコードの必要条件だが、十分条件ではない
- ・優れたコードは、可読性が高いコードである
  - ▶ 管理、変更、再利用が容易
  - ▶ 共同研究が容易
  - ▶ 透明性が高い
- ・コードの可読性 (readability) を高めよう！

# 可読性を高める(1)：コメント

- ・コメントをたくさん書くべし！
  - ▶ .Rmd ファイルならコードを説明する通常の文章（トピック4で説明する）
  - ▶ 他人のコードを読んでいるつもりになって、自分のコードの気になる点にコメントをつける
  - ▶ 「何をしたか」(what) だけでなく、「なぜそれをしたのか」(why) を書く
- ・実際にRで実行されるコードの分量は、コード全体の1/3未満になるのが普通：つまり、2/3以上はコメント！

# 可読性を高める(2)：スペースと改行

- ・スペースと改行を適切に利用する

- ・悪い例：

```
a<-(1+2)*4+5-8  
plot(x,y,xlim=c(1,10),ylim=c(-5,5),xlab="x-  
label",ylab="y-label",main="Title of Fig.")
```

- ・良い例：

```
a <- (1 + 2) * 4 + 5 - 8  
  
plot(x, y,  
      xlim = c(1,10), ylim = c(-5,5),  
      xlab = "x-label", ylab = "y-label",  
      main="Title of Fig.")
```

# 可読性を高める(3)：字下げによるブロック化

- 半角2文字（または4文字）の字下げで、コードをブロック化する

- 悪い例：

```
for(i in 1:N){  
  for(j in 1:K){  
    x[i,j]<-mean(rnorm(10))  
  }}
```

- 良い例：

```
for (i in 1:N) {      # loop for the rows  
  for (j in 1:K) {    # loop for the columns  
    x[i, j] <- mean(rnorm(10))  
  }  
}
```

\*RStudio には自動で字下げ(indent) する機能がある

# Topic 2 の課題

- 提出が必要な課題はなし
- まず、副読本の「基本的な操作」の章を読む
  - <http://www.jaysong.net/RBook/rbasic.html>
- 以下のページの内容を自分で実行する
  - <http://yukiyanai.github.io/jp/classes/stat2/contents/R/introduction-to-R.html>
    - ▶ 最初はコードをコピペしてRで実行し、解説と同じ結果が出るか確かめる
    - ▶ 次に、各コードの内容を、コードと実行結果を比較しながら1行ずつ確かめる
      - コードの中の数字を変えて、いろいろ試してみる

# 次回予告

3. 記述統計とデータの可視化