

2023/04/22 Tokyo.R #105 初心者セッション

# テーブルデータの取り扱い

---

がんばらないデータ加工(やさしめ)



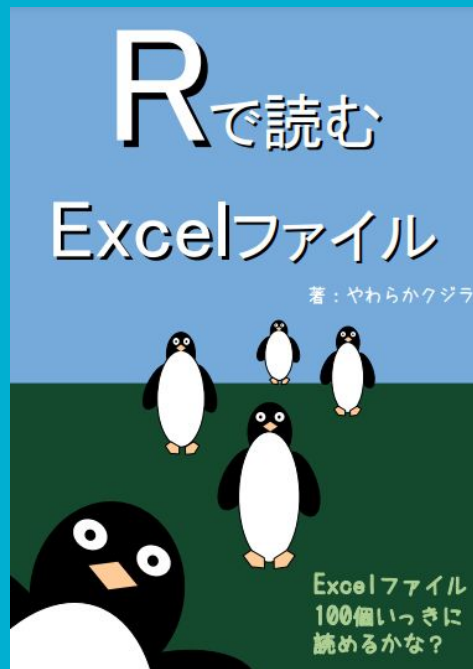
やわらかクジラ



:@matsuchiy

# 同人活動 (サークル名:ヤサイゼリー)

- 技術書典9にて頒布(0円)<sup>[1]</sup>
  - 技術書典13にてR4.2対応版公開(2022/9/16)
- RでのExcelファイルの読み書き
  - 1つ〜大量のxlsxファイル
  - csv × windowsの文字化け対処



[1] pdf: <https://techbookfest.org/product/4794168259903488?productVariantID=5913872206659584>.

# 同人活動


(サークル名:ヤサイゼリー)

- 技術書典12にて頒布(0円)<sup>[1]</sup>
  - 技術書典13にてR4.2対応版公開(2022/9/16)
- Rの基礎とdplyrの基本動詞を解説
  - くり返し作業を楽にしたい人に役立つ
    - ヘルパー関数, `rename_with()`, `across()`が分かる人には  
不要な本



[1] pdf: <https://techbookfest.org/product/5161487259664384?productVariantID=5672571053801472>.

# R for Data Science (2e) ; 略称 :r4ds

- 2023年4月時点では執筆途中
- 1stと比べ大幅に変更
  - %>% → |>
  - データ例に 



Hadley来日時の写真と直筆カード @ジュンク堂池袋本店プログラミングコーナー

R for Data Science  
(2e) 🔍 ☰

Welcome

Preface to the second edition

1 Introduction

Whole game

2 Data visualization

3 Workflow: basics

4 Data transformation

5 Workflow: code style

6 Data tidying

7 Workflow: scripts and projects

8 Data import

9 Workflow: getting help

R for Data Science (2e)

Welcome

This is the website for the work-in-progress 2nd edition of “**R for Data Science**”. This book will teach you how to do data science with R: You’ll learn how to get your data into R, get it into the most useful structure, transform it and visualize.

In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you’ll learn how to clean data and draw plots—and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You’ll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You’ll also learn how to manage cognitive resources to facilitate discoveries when wrangling, visualizing, and exploring data.

Table of contents

Welcome

Acknowledgements

Edit this page

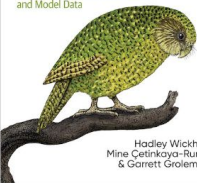
Report an issue

OREILLY

Second Edition

R for Data Science

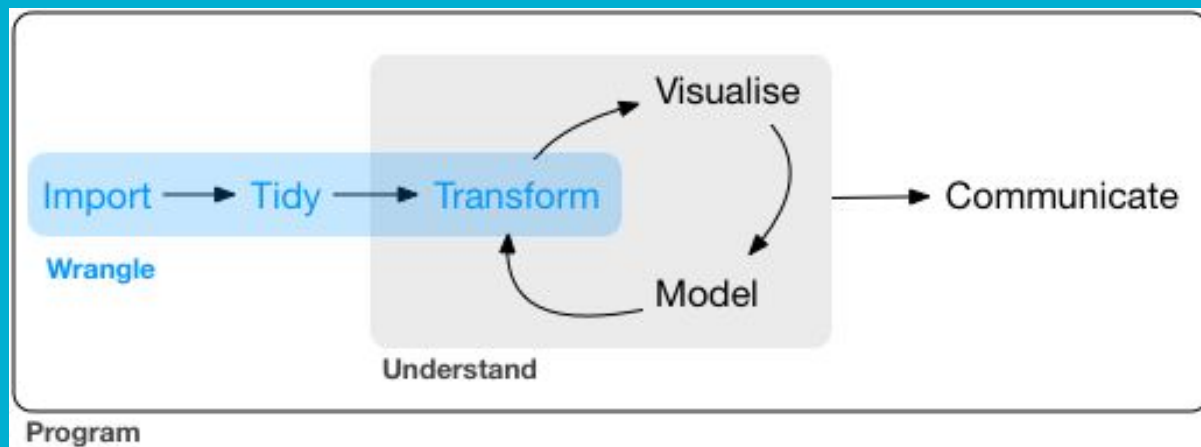
Import, Tidy, Transform, Visualize, and Model Data



Hadley Wickham,  
Mine Çetinkaya-Rundel  
& Garrett Grolemund

# データ加工とは

—

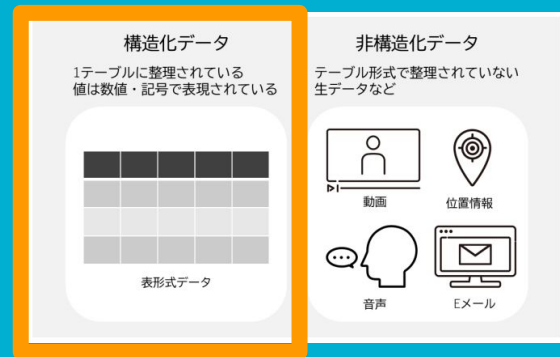


上図は1stより。2eだと, wrangleの網掛け図が見当たらない？

# データフレーム(data frame)

- **変数**(列:columns)と**オブザベーション**(行:rows)の集まった長方形
- データ加工を行うためのシンプルなデータの形
  - 構造化データ, 表形式(テーブル)データ

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007
Adelie	Torgersen	37.8	17.1	186	3300	NA	2007
Adelie	Torgersen	37.8	17.3	180	3700	NA	2007
Adelie	Torgersen	41.1	17.6	182	3200	female	2007

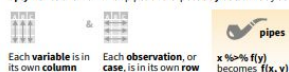


出典:総務省統計局「高等学校における「情報 II」のための  
データサイエンス・データ解析入門」

# やりたい作業はdplyrのチートシートで見つける

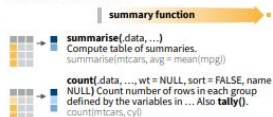
## Data transformation with dplyr : CHEAT SHEET

dplyr functions work with pipes and expect tidy data. In tidy data:



### Summarise Cases

Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).



### Group Cases

Use **group\_by**(data, ..., add = FALSE, drop = TRUE) to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.



Use **rowwise**(data, ...) to group data into individual rows. dplyr functions will compute results for each row. Also apply functions to list-columns. See tidy cheat sheet for list-column workflow.

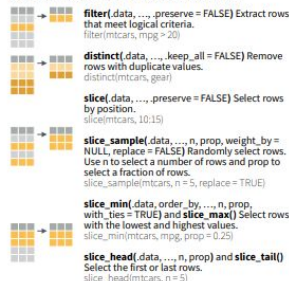


**ungroup**(x, ...) Returns ungrouped copy of table.  
ungroup(mtcars)

### Manipulate Cases

#### EXTRACT CASES

Row functions return a subset of rows as a new table.



**Logical and boolean operators to use with filter()**

==	<	<=	is.na()	%in%		xor()
!=	>	>=	is.na()	!	&	

See ?base:Logic and ?Comparison for help.

#### ARRANGE CASES



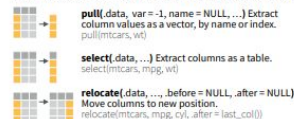
#### ADD CASES



### Manipulate Variables

#### EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.



Use these helpers with **select()** and **across()**



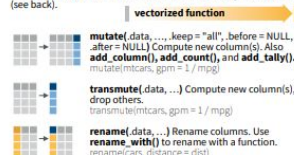
#### MANIPULATE MULTIPLE VARIABLES AT ONCE

**across**(cols, funs, ..., names = NULL) Summarise or mutate multiple columns in the same way.



#### MAKE NEW VARIABLES

Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).



# テーブルデータの取り扱いに慣れるために

---

- まず覚えたい{dplyr}の5つの動詞
  - 列(変数, カラム)を選ぶ : **select**
  - 列名を変更する : **rename**
  - 行(オブザベーション)を選ぶ: **filter**
  - 新しい列の作成 : **mutate**
  - 要約値を作る : **summarise**
- RStudio開いたら実行
  - `library(tidyverse)`





# RとPython


- R(dplyr)の方が大体シンプルな気がする



働き	R(dplyr)	Python(主にpandas)
列を選ぶ	<code>df  &gt; select(列1)</code>	<code>df[['列1']]</code>
列名を変更する	<code>df  &gt; rename(new = old)</code>	<code>df.rename(columns = {'old':'new'})</code>
行を選ぶ	<code>df  &gt; filter(列1 == 1)</code>	<code>df.query('列1 == 1')</code>
新しい列の作成	<code>df  &gt; mutate(new列1 = 列1)</code>	<code>df.assign(new列1 = df[列1])</code>
要約値を作る	<code>df  &gt; summarise(列1_mean = mean(列1, na.rm = TRUE))</code>	<code>df.agg({'列1':['mean']})</code>

# データ加工の練習開始



-  penguinsデータを読み込み
  - ボブさん@bob3bob3 によるtokyo.r発表資料([#99](#), [#101](#))

ケーキ  
写真



```
df <-  
palmerpenguins::penguins
```

<- Win/ Mac: Alt + - / Option + -

```
# データフレームの表示  
df
```

パッケージ名::関数など、の  
書き方で直接読みだせる

```
## # A tibble: 344 × 8  
##   species island   bill_le...1 bill_...2 flipp...3 body_...4 sex    year  
##   <fct>   <fct>     <dbl>     <dbl>    <int>    <int> <fct> <int>  
## 1 Adelie Torgersen    39.1      18.7     181     3750 male   2007  
## 2 Adelie Torgersen    39.5      17.4     186     3800 fema... 2007  
## 3 Adelie Torgersen    40.3       18     195     3250 fema... 2007  
## # ... with 341 more rows, and abbreviated variable names  
## #   'bill_length_mm', 'bill_depth_mm', 'flipper_length_mm',  
## #   'body_mass_g'
```

# 列(変数, カラム)を選ぶ: **select**



がんばり例: 延々と続く列, 気が遠くなるスクロール

	A	B	C	D	E
1	項目1	項目2	項目3	項目4	項目5
2					
3					
4					
5					

# 列(変数, カラム)を選ぶ: **select**

- `select()`の中に列名を入れるだけ
  - `()`内でtabキー押したら候補も出る



```
df |>
  select(bill_length_mm, bill_depth_mm)
```

```
## # A tibble: 344 × 2
##   bill_length_mm bill_depth_mm
##             <dbl>         <dbl>
## 1             39.1           18.7
## 2             39.5           17.4
## 3             40.3            18
## # ... with 341 more rows
```

データフレーム |>

適用する関数1(引数) |>

適用する関数2(引数) ...

|> (ベースパイプ/ネイティブパイプ)は  
R4.1で実装されて以降,  
%>% (マグリッターパイプ)に  
代わり使用が広まっている

Win/ Mac:

Ctrl + Shift + M / Cmd + Shift + M

# 列(変数, カラム)を選ぶ: **select**



がんばり例: フルの列名を並べていく

```
df |>
  select bill_length_mm, bill_depth_mm)
```

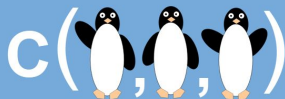
```
## # A tibble: 344 × 2
##   bill_length_mm bill_depth_mm
##           <dbl>         <dbl>
## 1             39.1             18.7
## 2             39.5             17.4
## 3             40.3              18
## # ... with 341 more rows
```

Q. 同じ文字が入ってる列名は省略できませんか？

A. **ヘルパー関数**を使って効率化できます。  
続きは→

がんばらない  
データ加工

Rによるくり返し  
作業入門 前編



著: やわらかクジラ

# 列名を変更する: **rename**



がんばり例: 手動でひたすら入力, 置換で意図しないミス

	A	B	C	D	E
1	Var_1	Var_2	Var_3	Var_4	Var_5
2					
3					

# 列名を変更する: **rename**

- `rename()`の中に「新しい列名 = 古い列名」と書くだけ
  - ただしこれだけなら `select()`でもできる



```
df |>
  rename(くちばしの長さ = bill_length_mm)
```

```
## # A tibble: 344 × 8
##   species island   くちば...1 bill_d...2 flipp...3 body_...4 sex    year
##   <fct>   <fct>         <dbl>     <dbl>     <int>     <int> <fct> <int>
## 1 Adelie  Torgersen      39.1      18.7      181      3750 male   2007
## 2 Adelie  Torgersen      39.5      17.4      186      3800 fema... 2007
## 3 Adelie  Torgersen      40.3       18       195      3250 fema... 2007
## # ... with 341 more rows, and abbreviated variable names
## #   1くちばしの長さ, 2bill_depth_mm, 3flipper_length_mm,
## #   4body_mass_g
```

# 列名を変更する: **rename**



がんばり例: 同じパターンの変更を何度も書いていく

```
df |>
  rename(くちばし_length_mm = bill_length_mm,
         くちばし_depth_mm = bill_depth_mm)
```

```
## # A tibble: 344 × 8
##   species island   くちば...1 くちば...2 flipp...3 bo
##   <fct>    <fct>      <dbl>    <dbl>    <int>
## 1 Adelie  Torgersen      39.1      18.7     181
## 2 Adelie  Torgersen      39.5      17.4     186
## 3 Adelie  Torgersen      40.3      18      195
## # ... with 341 more rows, and abbreviated variables:
## #   1くちばし_length_mm, 2くちばし_depth_mm, 3fl
## #   4body_mass_g
```

Q. 他のたくさんの列名も一括で変えたいんですけど？

A. **rename\_with()**を使って効率化できます。

続きは→

がんばらない  
データ加工

Rによるくり返し  
作業入門 前編



著: やわらかクジラ



# 行 (オブザベーション) を選ぶ: **filter**

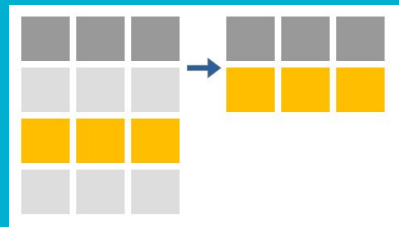


がんばり例: 延々と続く行, 気が遠くなるスクロール

	A	B
1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
10		10
11		11
12		12

# 行 (オブザベーション) を選ぶ: **filter**

- filter()内に式を書いて指定した行だけにする



```
df |>
  filter(species == "Gentoo")
```

TRUE or FALSEを返すもの。他にも  
<, <=, >, >=, !=  
が使える

```
## # A tibble: 124 × 8
##   species island bill_length...1 bill_...2 flipp...3 body_...4 sex    year
##   <fct>    <fct>         <dbl>    <dbl>    <int>    <int> <fct> <int>
## 1 Gentoo  Biscoe           46.1     13.2     211     4500 fema... 2007
## 2 Gentoo  Biscoe           50      16.3     230     5700 male   2007
## 3 Gentoo  Biscoe           48.7     14.1     210     4450 fema... 2007
## # ... with 121 more rows, and abbreviated variable names
## #   1bill_length_mm, 2bill_depth_mm, 3flipper_length_mm,
## #   4body_mass_g
```

# 行(オブザベーション)を選ぶ: filter

- うろ覚えでも必要な行を表示できる

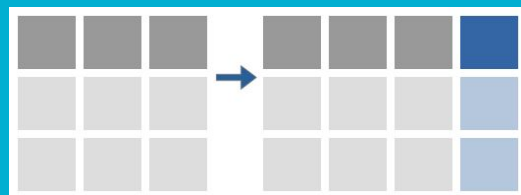
```
df |>
  filter(str_detect(species, "Ade"))
```

stringr::str\_系の関数は  
" "の中が正規表現



```
## # A tibble: 152 × 8
##   species island    bill_le...1 bill_...2 flipp...3 body_...4 sex    year
##   <fct>    <fct>      <dbl>    <dbl>    <int>    <int> <fct> <int>
## 1 Adelie  Torgersen    39.1     18.7     181     3750 male   2007
## 2 Adelie  Torgersen    39.5     17.4     186     3800 fema... 2007
## 3 Adelie  Torgersen    40.3     18      195     3250 fema... 2007
## # ... with 149 more rows, and abbreviated variable names
## #   1bill_length_mm, 2bill_depth_mm, 3flipper_length_mm,
## #   4body_mass_g
```

# 新しい列の作成: **mutate**



- ある列に関数や計算式を適用して新しい列を作成

```
df |> mutate(mean_blmm = mean(bill_length_mm, na.rm = TRUE), # bill_length_mmの平均値列を作成
            dif_blmm_mean = bill_length_mm - mean_blmm,      # 各個体で全体平均値との差分作成
            .keep = "used")                                   # 使用した列のみ残す
```

```
## # A tibble: 344 × 3
##   bill_length_mm mean_blmm dif_blmm_mean
##           <dbl>     <dbl>         <dbl>
## 1             39.1      43.9          -4.82
## 2             39.5      43.9          -4.42
## 3             40.3      43.9          -3.62
## # ... with 341 more rows
```

# 新しい列の作成: **mutate**



がんばり例: 対象となる個々の列(変数)にすべて同じ処理を書く

```
df |>
  mutate(species_c = as.character(species),
         island_c = as.character(island),
         .keep = "used")
```

```
## # A tibble: 344 × 4
##   species island   species_c island_c
##   <fct>   <fct>     <chr>      <chr>
## 1 Adelie  Torgersen Adelie    Torgersen
## 2 Adelie  Torgersen Adelie    Torgersen
## 3 Adelie  Torgersen Adelie    Torgersen
## # ... with 341 more rows
```

Q. 他のたくさんの列にも同じ処理したいんですけど？

A. **across()**を使って効率化できます。  
続きは→

がんばらない  
データ加工

Rによるくり返し  
作業入門 前編



著: やわらかクジラ

# 要約値を作る: summarise

- 関数を列に適用した結果をデータフレームで返す



```
df |>
```

new = 関数を列に適用

```
  summarise(blm_mean = mean(bill_length_mm, na.rm = TRUE),  
            blm_sd   = sd(bill_length_mm, na.rm = TRUE))
```

```
## # A tibble: 1 × 2  
##   blm_mean blm_sd  
##   <dbl>   <dbl>  
## 1    43.9    5.46
```

# 要約値を作る: summarise



がんばり例: 各列(変数)にすべての関数を適用

```
df |>
  summarise(blm_mean = mean(bill_length_mm, na.rm = TRUE),
            blm_sd   = sd(bill_length_mm, na.rm = TRUE),
            bdm_mean = mean(bill_depth_mm, na.rm = TRUE),
            bdm_sd   = sd(bill_depth_mm, na.rm = TRUE))
```

```
## # A tibble: 1 × 4
##   blm_mean blm_sd bdm_mean bdm_sd
##   <dbl>   <dbl>   <dbl>   <dbl>
## 1    43.9    5.46    17.2    1.97
```

Q. 他のたくさんの列にも同じ処理したいんですけど？

A. **across()**を使って効率化できます。

続きは→

がんばらない  
データ加工

Rによるくり返し  
作業入門

前編



著: やわらかクジラ

# 他にもやりたい作業いろいろ

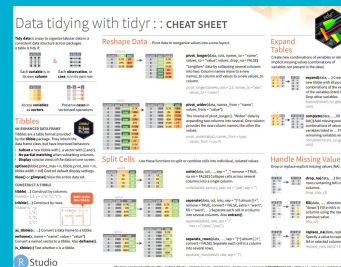
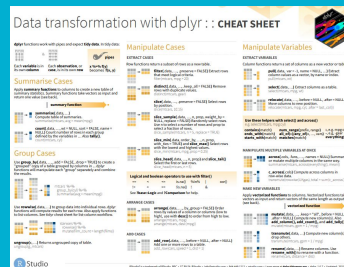
- テーブル同士をキーとなる列の値や列名で連結したい
  - 横: `left_join()`, `inner_join()` など
  - 縦: `bind_rows()` など
- テーブルを横 $\longleftrightarrow$ 縦にしたい (wide $\longleftrightarrow$ long)
  - `tidyr::pivot_longer()`, `tidyr::pivot_wider()`

ID	2022	2023	ID	year	score
1	50	100	1	2022	50
2	70	80	2	2022	70
			1	2023	100
			2	2023	80

X	ID	A	B	+	Y	ID	A	C
	1	a	あ			1	a	イ
	2	b	い			2	b	ロ
	3	c	う			4	d	ニ

	ID	A	B
X	1	a	あ
	2	b	い
	ID	A	B
Y	3	c	う
	4	d	え

シートで大体分かる

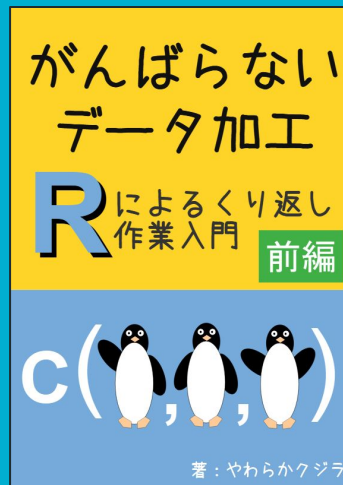




# まとめ

---

- RStudio起動したらまずlibrary(tidyverse)
- dplyrの基本動詞に慣れる
  - select(), rename(), filter(), mutate(), summarise()
- 同じ処理のくり返しが嫌になったら拙同人誌へ→



---

# Enjoy!

