

はじめてのR講座REVIVAL

久我, 立
九州大学大学院システム生命科学府

<https://hdl.handle.net/2324/4492892>

出版情報 : 2021-09-24. 九州大学附属図書館
バージョン :
権利関係 :



はじめての

R

事後
配布

講座

REVIVAL

所属 システム生命科学府一貫制博士5年
(理学部 生物学科 卒業)

研究室 生態科学研究室

研究内容 行動生態学 (バッタの逃避行動)



Rはローテク(?)な実験中心の学生でも使える！

Rを使ったデータ解析を体験

主な内容

- ・ Rって何なの？
- ・ Rでは何ができるの？
- ・ どうやって使うの？
- ・ Rをさらに使いこなす

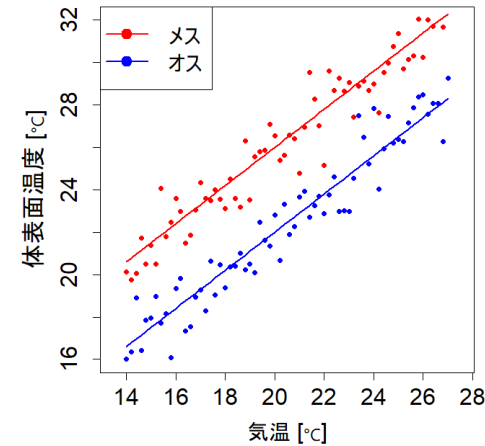
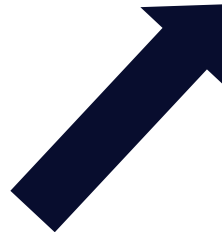
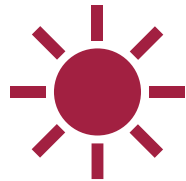
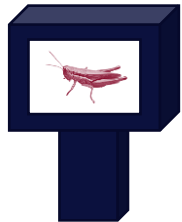


コードをコピーしながら
実際に使ってみましょう！

Step. 1

Rって何なの？

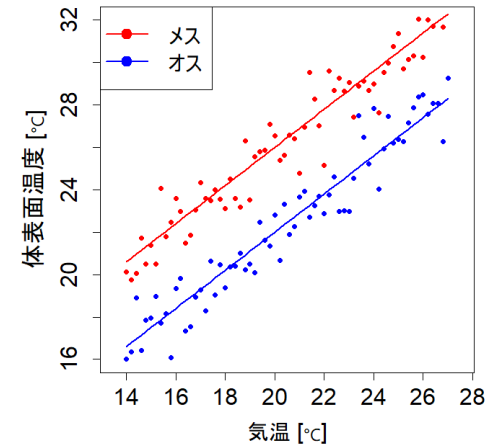
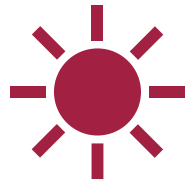
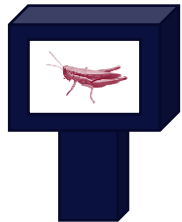
バッタの体表面温度は
気温と関係があるのか？



おしまい。

様々な気温の下で
体表面温度を測定

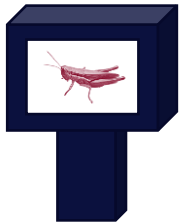
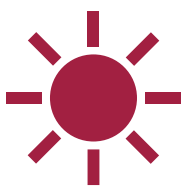
バッタの体表面温度は
気温と関係があるのか？



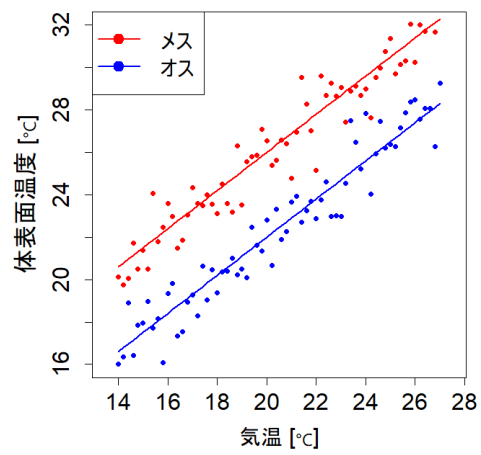
おしい。

様々な気温の下で
体表面温度を測定

バッタの体表面温度は
気温と関係があるのか？



様々な気温の下で
体表面温度を測定



データの統計解析

データの図化



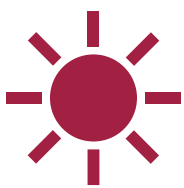
データの整理



日時	気温	体表面温度	性
2020/7/3	24.5	27.8	オス

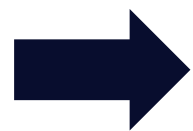
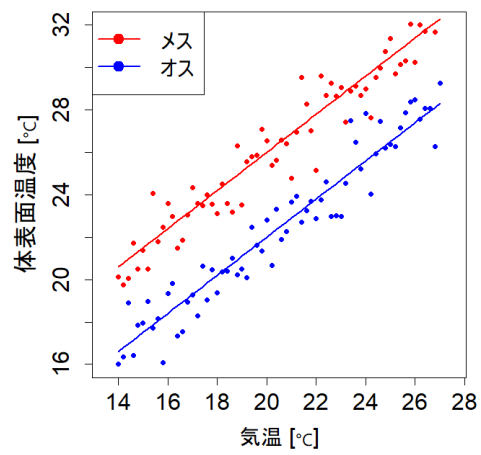
データのデジタル化

バッタの体表面温度は
気温と関係があるのか？



様々な気温の下で
体表面温度を測定

さらに...



データの統計解析

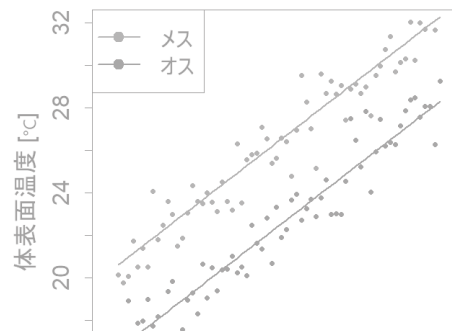
データの図化

データの整理

日時	気温	体表面温度	性
2020/7/3	24.5	27.8	オス

データのデジタル化

バッタの体表面温度は
気温と関係があるのか？



無限解析地獄

データの統計解析



様々な気温の下で
体表面温度を測定



日時	気温	体表面温度	性
2020/7/3	24.5	27.8	オス

データのデジタル化

https://upload.wikimedia.org/wikipedia/commons/6/67/Horse-Face_of_the_Underworld.jpg



あなた

データ

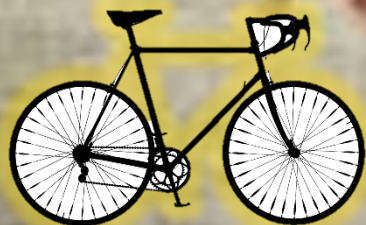
https://upload.wikimedia.org/wikipedia/commons/6/67/Horse-Face_of_the_Underworld.jpg

R = 自転車のイメージです。
あるとかなり便利ですが、
あるからといって地獄から
抜け出せるとは限りません…

あなた

データ

R





統計解析と作図に特化した フリーソフト

R言語を用いたプログラミング

Python
機械学習



統計解析のパッケージが充実している

多くの研究者が使用している

[PDF] [R: A language and environment for statistical computing](#)

[RC Team](#) - 2013 - [r.meteo.uni.wroc.pl](#)

This document describes basic features of dplR by following the initial steps that an analyst might follow when working with a new tree-ring data set. The vignette starts with reading in ring widths and plotting them. We describe a few of the available methods for detrending ...

☆ 99 引用元 233004 関連記事 全 15 ページョン 88

Rを引用した文献

23万超え (2021.7.19現在)

Excelの”よいところ”

みんながよく使うデータ解析を簡単に実施できる

Excelの”ダメなところ” → **R**におまかせ!

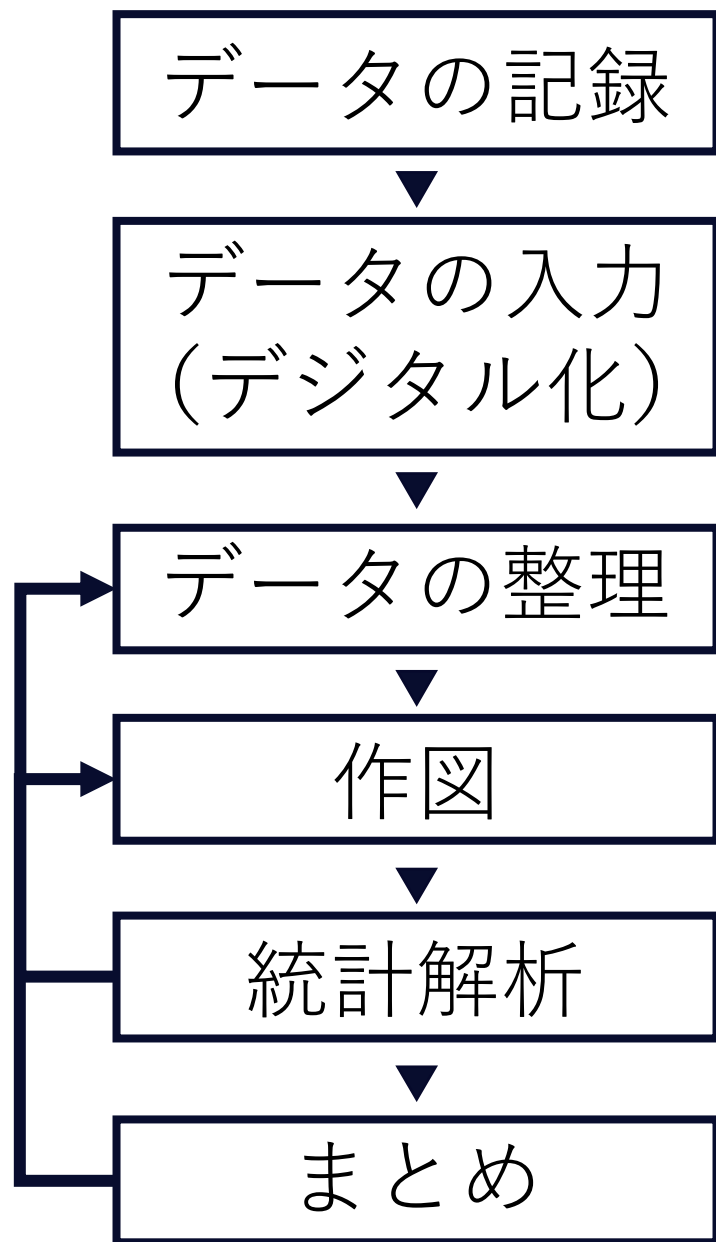
みんながよく使うデータ以外の解析が難しい

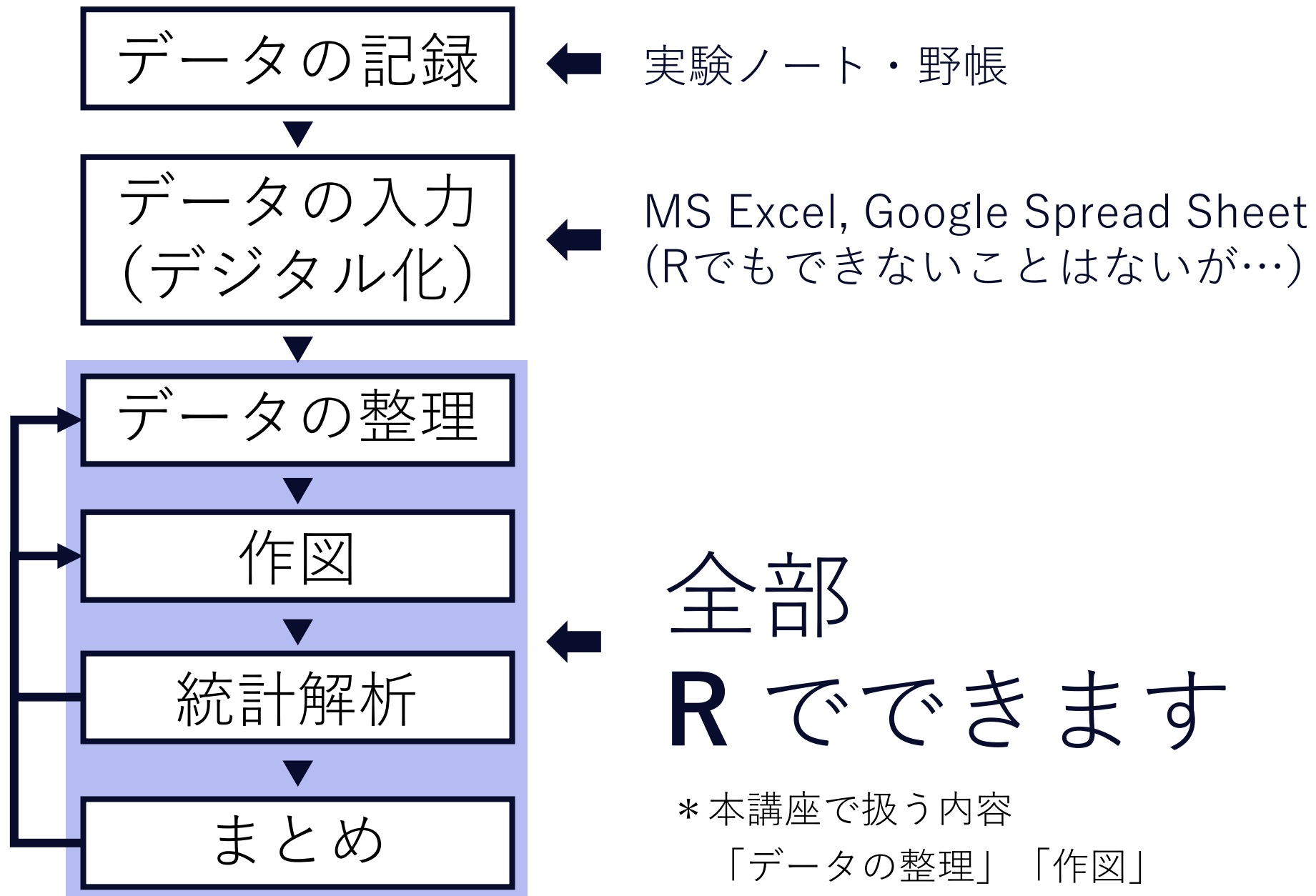
データそのものをどういじったのか履歴が残らない
(データ解析の再現性が低い)

データに変更が生じた際に
一から作業をやり直す必要がある

Step. 2

**Rでは
何ができるの？**





Step. 3

どうやって
使うの？

まずは Excel でデータ入力 列(縦)

行
(横)

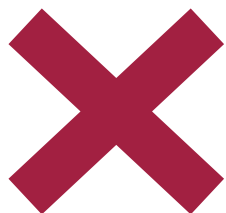
	A	B	C	D	E
1	Name	Subject	Score	1行目の変数名	
2	Tanaka	Math	85	2行目以降に データ(値)	
3	Sato	Math	74		
4	Otsu	Math	92		
5					

データ入力3原則

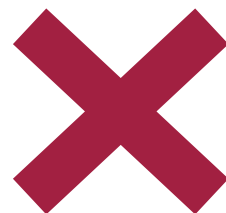
なるべく半角で入力

1列 = 1変数 1行 = 1サンプル 1セル = 1つの値

この条件を満たす構造のデータをRではデータフレームと呼ぶ



	A	B	C	D	E
1		左上から入力していない			
2					
3		Name	Subject	Score	
4		Tanaka	Math	85	
5		Sato	Math	74	
6		Otsu	Math	92	
7					



	A	B	C	D	E
1	Name	Subject	Score		
2	Tanaka	Math	85		
3	Sato	Math	74		
4	Otsu	Math	92		
5					
6		平均	84		
7		1つの行列以外の要素がある			



	A	B	C	D	E
1	Name	Subject	Score	← 列名	
2	Tanaka	Math	85		
3	Sato	Math	74		
4	Otsu	Math	92		
5					
6	1つの行列になっている				
7					

欠損データは NA と入力

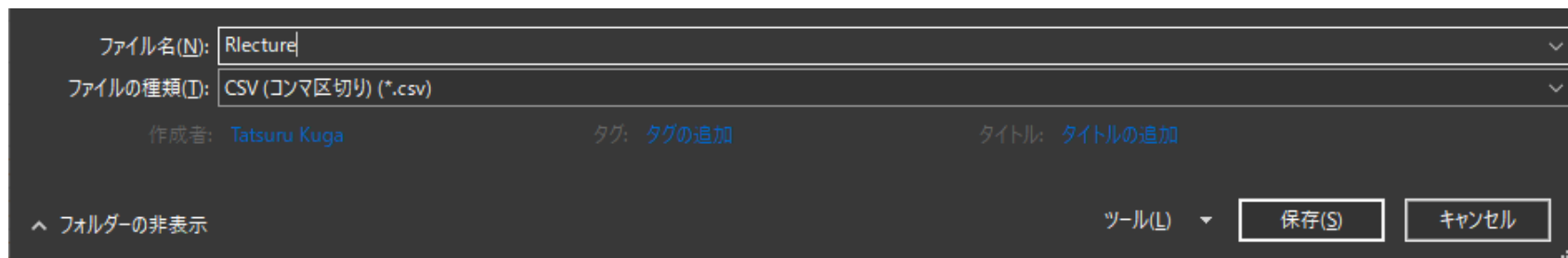
	A	B	C
1	Name	Subject	Score
2	Tanaka	Math	85
3	Sato	Math	NA ←
4	Otsu	Math	92

* 0 ではないため0と入力しない

* 入力忘れではないため空欄にしない

ファイル > 名前をつけて保存

ファイルの種類：CSV（コンマ区切り）



* windowsユーザーはUTF-8のない方で保存する！
(以下はダメ)



ここから Rを使います

表記の見方

皆さんに実際に入力してほしいコードです。

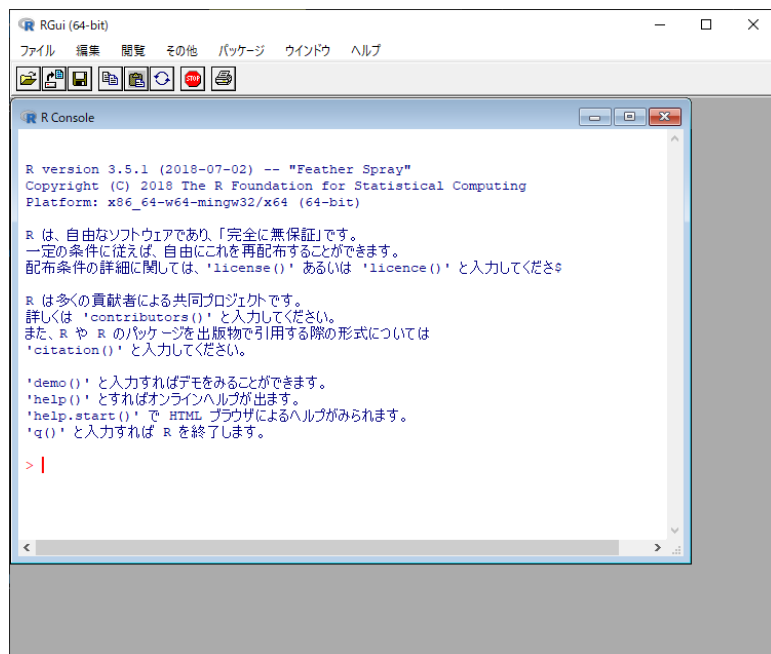
入力例

コードをRに入力した場合の見え方です。

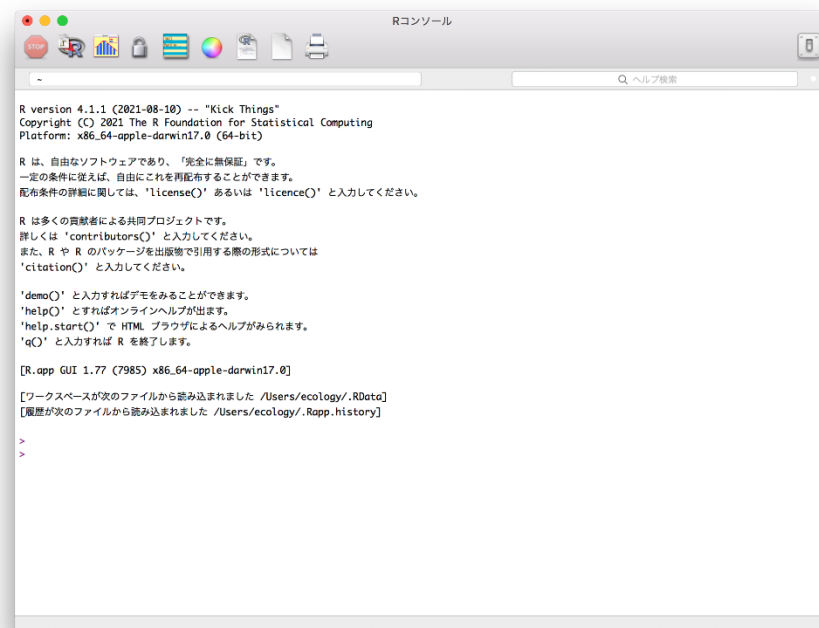
コードを一般化したものです。



デスクトップの「R」アイコンを
ダブルクリックで起動

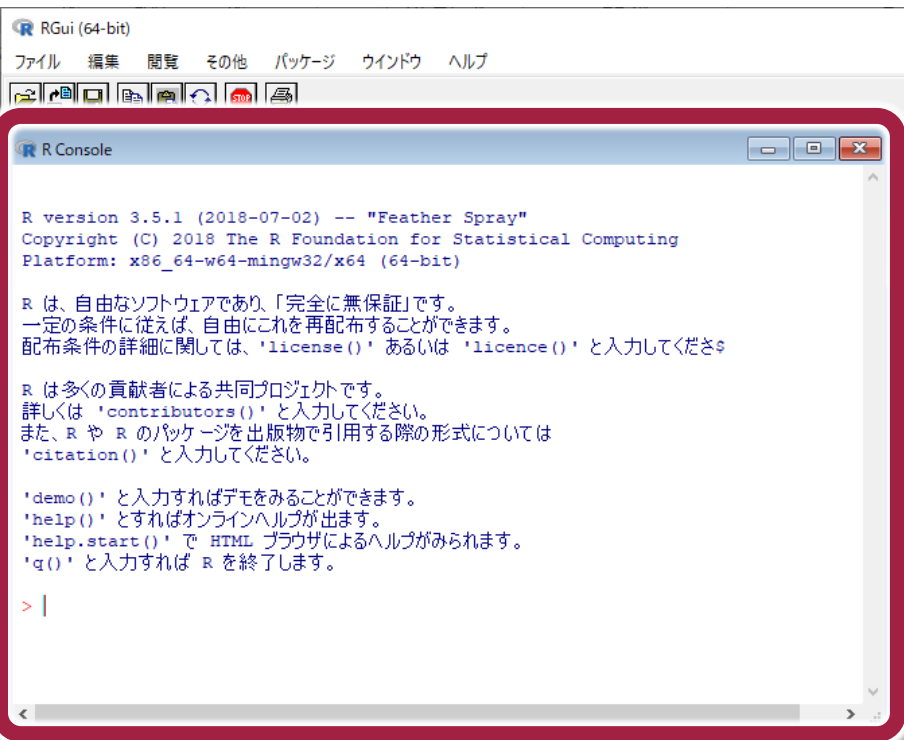


windowsの場合



Macの場合

コンソール画面に入力



```
RGui (64-bit)
ファイル 編集 閲覧 その他 パッケージ ウィンドウ ヘルプ

R Console

R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してください

R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' と入力してください。
また、R や R のパッケージを出版物で引用する際の形式については
'citation()' と入力してください。

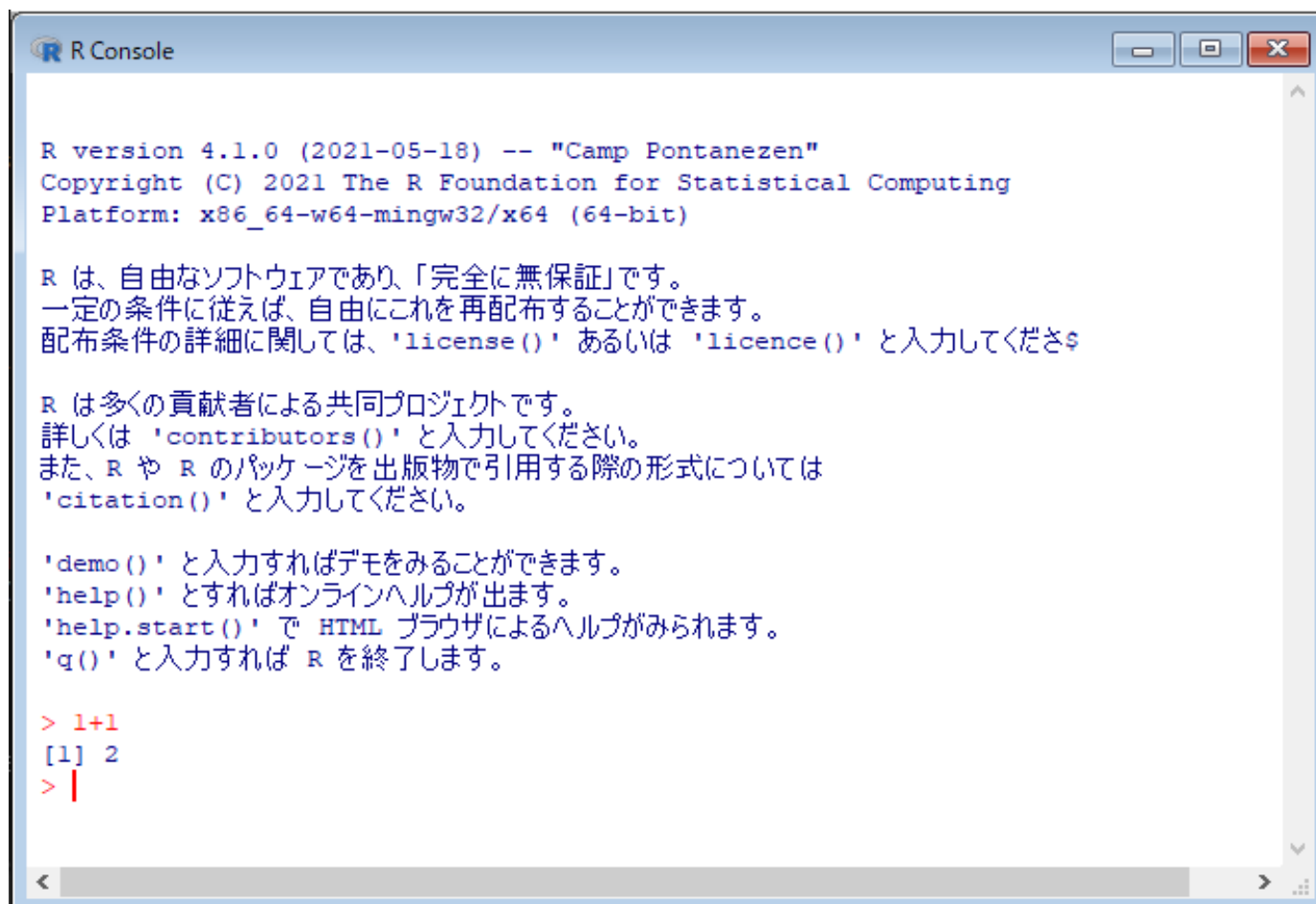
'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます。
'q()' と入力すれば R を終了します。

> |
```

これが
コンソール画面
(>の横に書かれた
内容が実行される)

試しに $1+1$ と入力して Enter

コンソール画面の見方



```
R Console

R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してください$

R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' と入力してください。
また、R や R のパッケージを出版物で引用する際の形式については
'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます。
'q()' と入力すれば R を終了します。

> 1+1
[1] 2
> |
```

赤字 が命令

青字 が命令の実行結果

Rへのcsvデータの取り込み

```
mydata <- read.csv("_____/battadata.csv")
```

* 下線部に自身のファイルのパスを書いて, Rに入力してください

入力例 `> mydata<-read.csv("C:/Users/tgoni/Desktop/battadata.csv")`

任意のオブジェクト名 <- read.csv("ファイルのパス/ファイル名.csv")

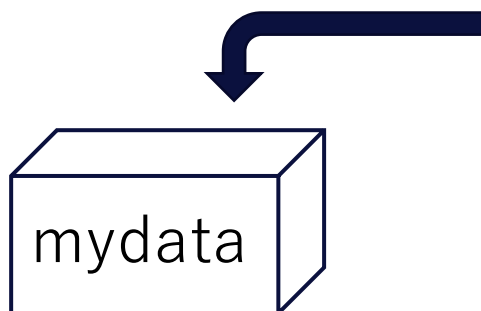
②

③

①

- ① ファイルを読み込んで ② 任意の名前のオブジェクトに
③ 代入する

オブジェクト = データを入れる「箱」



	A	B	C	D	E
1	Name	Subject	Score		
2	Tanaka	Math	85		
3	Sato	Math	74		
4	Otsu	Math	92		
5					
6		平均	84		
7					

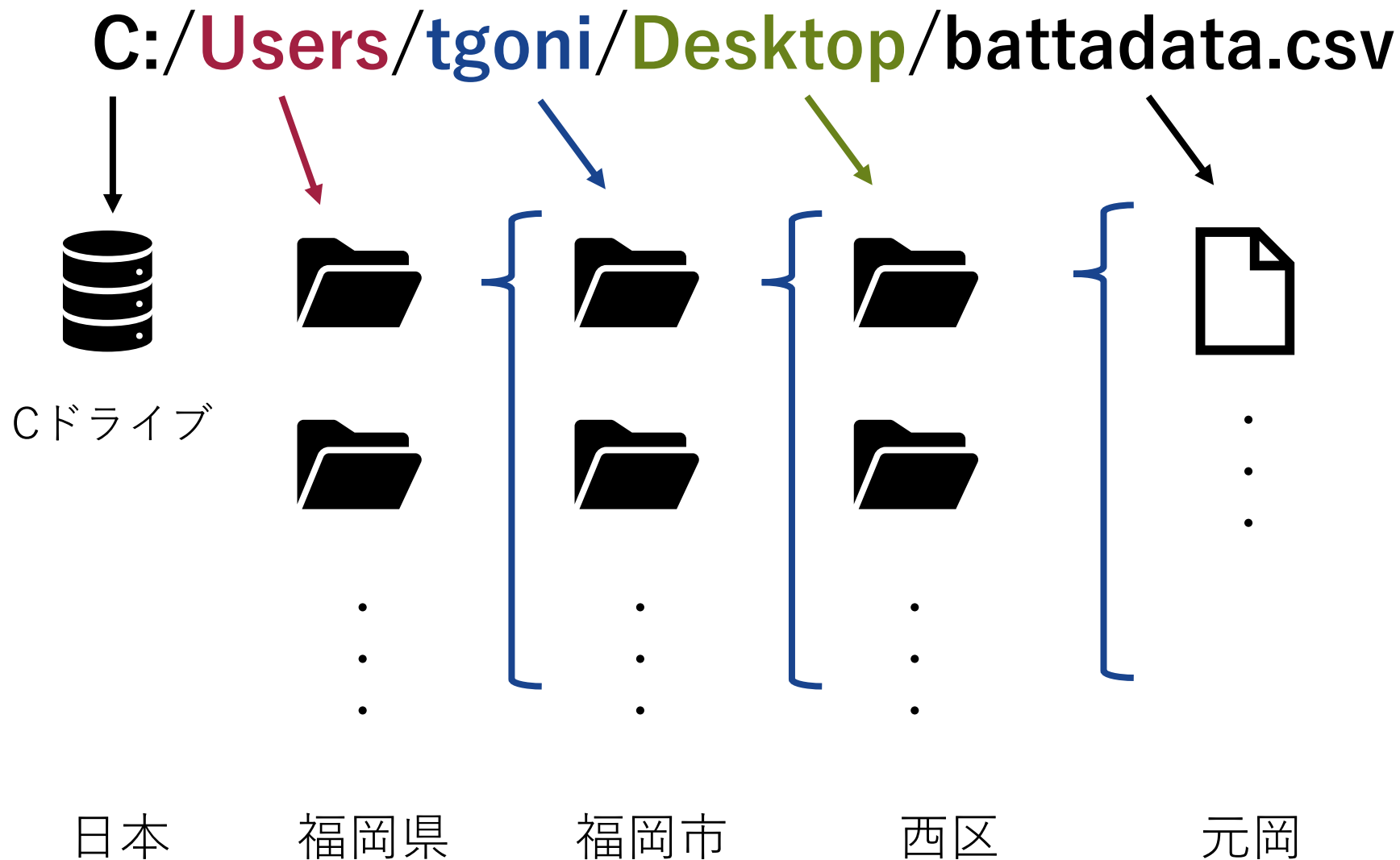
データ

TIPS

* 文字列は “ ” で囲む

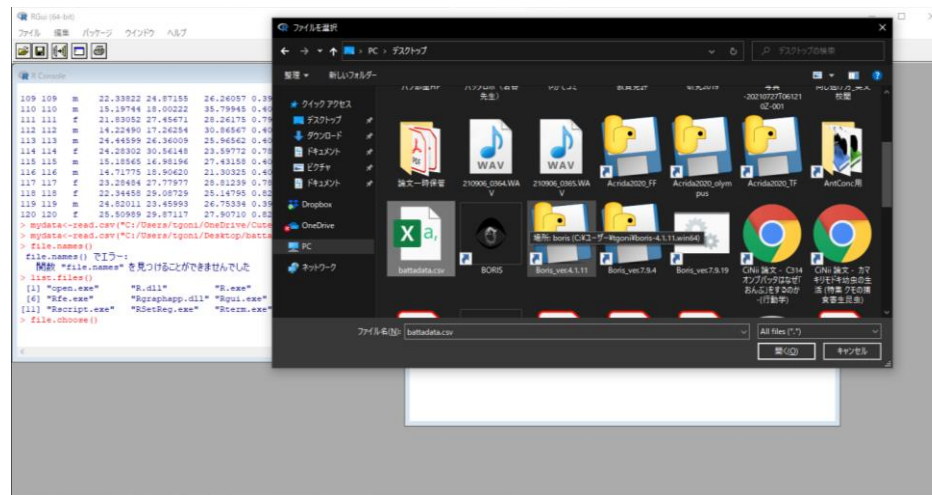
* 記号 <- は「代入」の意味

ファイルの住所のようなものです。



file.choose()

実行後、目的のファイル（.csv）を選んで「開く」



入力例

```
> file.choose()  
[1] "C:\\Users\\tgoni\\Desktop\\battadata.csv"
```

入力例

この部分をコピーしてread.csv()に貼り付け！

```
> mydata<-read.csv("C:\\Users\\tgoni\\Desktop\\battadata.csv")
```

* \\ は / にしても同じ（パスに限る）

オブジェクト名を入力すれば中身が表示される

入力例

```
> mydata
  id sex temperature bodytemp bodylength  weight  place
1   1  f   26.79358  29.98917   28.33527 0.8033418 maedashi
2   2  f   17.54348  25.05597   32.60819 0.8248300 maedashi
3   3  m   16.58518  18.45797   27.88323 0.4002156 hakozaqi
4   4  m   14.18262  15.80969   24.43933 0.3988926 hakozaqi
5   5  m   24.29460  26.51175   22.69562 0.3987375      ito
6   6  m   26.82988  28.79264   26.85507 0.3988612 hakozaqi
7   7  m   23.41778  26.25568   31.24154 0.3963641 maedashi
8   8  f   18.71212  24.28951   30.73882 0.8111704 hakozaqi
9   9  f   26.15488  30.64739   20.62201 0.8338814 maedashi
10  10  f   19.70098  24.30618   33.94342 0.8082500 hakozaqi
11  11  m   22.12780  24.62894   17.60382 0.4016618 hakozaqi
12  12  m   15.13878  16.46915   27.81868 0.4015646      ito
13  13  f   17.16076  23.67592   32.56590 0.8082212 hakozaqi
14  14  f   18.92568  24.93928   23.71431 0.8117349 hakozaqi
15  15  f   22.77808  27.51790   21.17245 0.8026716 hakozaqi
16  16  m   17.50362  19.93808   26.36772 0.4005058 maedashi
17  17  f   25.67111  31.43059   25.79896 0.8049550 hakozaqi
18  18  m   16.00683  18.24887   22.57396 0.3994941 hakozaqi
19  19  m   18.87423  20.27894   28.27413 0.3959700      ito
20  20  m   21.34785  23.96484   25.58410 0.4016459 maedashi
```

データを読み込んだ後は必ず中身を確認しておく！
(解析後に中身が違うことに気づくとツライ)

abcde

例：mydata

オブジェクト

abcdeという箱
(中に数字や文字が入っている)

abcde()

例：read.csv()

関数

abcdeという名前で定義した
一連の処理を実施

“abcde”

例：“C:/Users/tgoni/Desktop/battadata.csv”

文字

abcdeという文字

データの入力

Excelで作成し、csv形式で保存

1列=1変数、1行=1サンプル、1セル=1つの値

データの読み込み

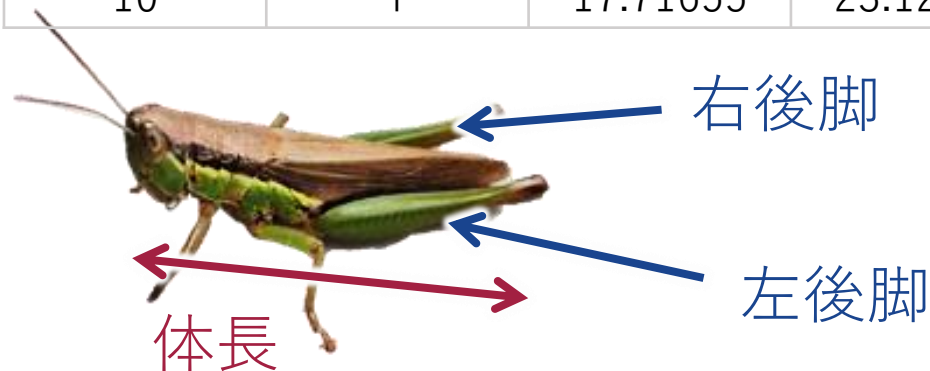
`read.csv()`を使ってcsvデータを読み込む

パスが分からないときは `file.choose()` で調べる

データフレーム battadata.csv の中身

バッタの体温を各キャンパスで記録したデータです。

ID	性	気温	体表面温度	体長	体重	採集場所
id	sei	temperature	bodytemp	bodylength	weight	place
1	f	19.66227	26.32604	29.35272	0.824753	maedashi
2	m	16.94141	19.05028	18.58453	0.401179	hakozaki
3	m	23.83568	24.29131	20.58872	0.39919	ito
4	m	24.88971	27.15717	20.23409	0.401533	ito
5	m	23.11532	24.92456	20.03364	0.399128	maedashi
6	m	15.90721	18.44454	20.18955	0.401698	maedashi
7	f	17.8181	24.37523	30.1241	0.779901	maedashi
8	m	18.70666	21.04032	17.74339	0.401306	ito
9	f	17.36528	22.56657	30.68803	0.811133	hakozaki
10	f	17.71655	23.12332	28.68992	0.790906	maedashi



今回は1変数～2変数のデータを扱っていきます。

入力したデータを1列ずつチェックする

ID	性	気温	体表面温度	体長	体重	採集場所
id	sei	temperature	bodytemp	bodylength	weight	place
1	f	19.66227	26.32604	29.35272	0.824753	maedashi
2	m	16.94141	19.05028	18.58453	0.401179	hakozaki
3	m	23.83568	24.29131	20.58872	0.39919	ito
4	m	24.88971	27.15717	20.23409	0.401533	ito
5	m	23.11532	24.92456	20.03364	0.399128	maedashi
6	m	15.90721	18.44454	20.18955	0.401698	maedashi
7	f	17.8181	24.37523	30.1241	0.779901	maedashi
8	m	18.70666	21.04032	17.74339	0.401306	ito
9	f	17.36528	22.56657	30.68803	0.811133	hakozaki
10	f	17.71655	23.12332	28.68992	0.790906	maedashi
カテゴリー		数値	数値	数値	数値	カテゴリー

数値 → summary()で要約 + ヒストグラムで視覚化

カテゴリー → summary()で要約

mydata\$bodylength

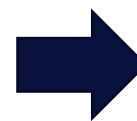
入力例

```
> mydata$bodylength
[1] 28.33527 32.60819 27.88323 24.43933 22.69562 26.85507 31.24154 30.73882
[9] 20.62201 33.94342 17.60382 27.81868 32.56590 23.71431 21.17245 26.36772
[17] 25.79896 22.57396 28.27413 25.58410 23.31778 22.64082 32.54974 16.79654
[25] 28.12095 25.93147 24.64630 24.83070 37.01561 27.62770 25.35283 26.06685
[33] 15.62503 33.11657 21.38008 22.65300 22.48243 20.34304 19.44260 38.14324
[41] 33.18955 25.83104 26.62024 22.41717 19.33950 11.53776 40.39468 33.29256
```

mydata

id	sei	temperature	bodytemp	bodylength	weight	place
1	f	19.66227	26.32604	29.35272	0.824753	maedashi
2	m	16.94141	19.05028	18.58453	0.401179	hakozaiki
3	m	23.83568	24.29131	20.58872	0.39919	ito
4	m	24.88971	27.15717	20.23409	0.401533	ito
5	m	23.11532	24.92456	20.03364	0.399128	maedashi
6	m	15.90721	18.44454	20.18955	0.401698	maedashi
7	f	17.8181	24.37523	30.1241	0.779901	maedashi
8	m	18.70666	21.04032	17.74339	0.401306	ito
9	f	17.36528	22.56657	30.68803	0.811133	hakozaiki
10	f	17.71655	23.12332	28.68992	0.790906	maedashi

mydata\$bodylength



bodylength
29.35272
18.58453
20.58872
20.23409
20.03364
20.18955
30.1241
17.74339
30.68803
28.68992

データフレームのオブジェクト \$ 列名

オブジェクトから列名の列を抽出
(ベクトルとして出力される)

* 記号 \$ は「~の」の
イメージ

抽出したデータをオブジェクトに格納する

列を抽出するだけ

```
mydata$place
```

place
maedashi
hakovaki
ito
ito
maedashi
maedashi
maedashi
ito
hakovaki
maedashi

id	sei	temperature	bodytemp	bodylength	weight	place
1	f	19.66227	26.32604	29.35272	0.824753	maedashi
2	m	16.94141	19.05028	18.58453	0.401179	hakovaki
3	m	23.83568	24.29131	20.58872	0.39919	ito
4	m	24.88971	27.15717	20.23409	0.401533	ito
5	m	23.11532	24.92456	20.03364	0.399128	maedashi
6	m	15.90721	18.44454	20.18955	0.401698	maedashi
7	f	17.8181	24.37523	30.1241	0.779901	maedashi
8	m	18.70666	21.04032	17.74339	0.401306	ito
9	f	17.36528	22.56657	30.68803	0.811133	hakovaki
10	f	17.71655	23.12332	28.68992	0.790906	maedashi

列を抽出して、名前をつける

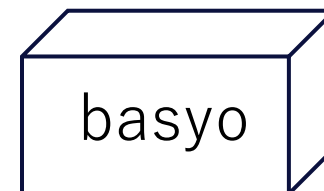
```
basyo<-mydata$place
```

代入記号

オブジェクト名

place
maedashi
hakovaki
ito
ito
maedashi
maedashi
maedashi
ito
hakovaki
maedashi

id	sei	temperature	bodytemp	bodylength	weight	place
1	f	19.66227	26.32604	29.35272	0.824753	maedashi
2	m	16.94141	19.05028	18.58453	0.401179	hakovaki
3	m	23.83568	24.29131	20.58872	0.39919	ito
4	m	24.88971	27.15717	20.23409	0.401533	ito
5	m	23.11532	24.92456	20.03364	0.399128	maedashi
6	m	15.90721	18.44454	20.18955	0.401698	maedashi
7	f	17.8181	24.37523	30.1241	0.779901	maedashi
8	m	18.70666	21.04032	17.74339	0.401306	ito
9	f	17.36528	22.56657	30.68803	0.811133	hakovaki
10	f	17.71655	23.12332	28.68992	0.790906	maedashi



```
summary(mydata$bodylength)
```

入力例

```
> summary(mydata$bodylength)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
17.20	20.21	25.06	25.26	30.37	34.82

最小値

第一四分位数

中央値

平均値

第三四分位数

最大値

```
summary(数値ベクトル)
```

数値のベクトルデータの要約情報を表示

ベクトル：一次元のデータ

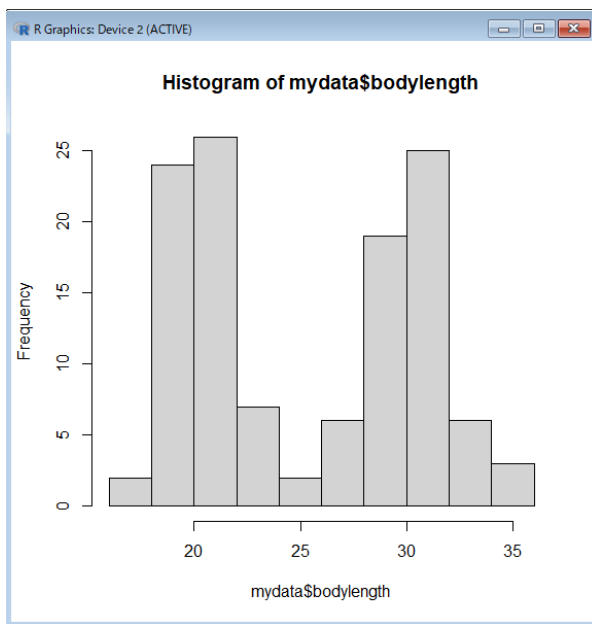
行列：二次元のデータ（ベクトルの集まり）

 $(1 \ 2 \ 3)$ $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ **Tips**

* 一列の行列も存在するため、注意

```
hist(mydata$bodylength)
```

入力例 `> hist(mydata$bodylength)`



外れ値はないか？

2 峰分布

→異なるデータが混じっていないか？

使う予定の検定の仮定に矛盾しないか

```
hist (数値のベクトル)
```

数値のベクトルのヒストグラムを作成

subset() を使って条件を満たすデータ（行）を取り出す

```
mydata_male<-subset(mydata,sei=="m")  
mydata_female<-subset(mydata,sei=="f")
```

入力例

```
> mydata_male<-subset(mydata,sei=="m")  
> mydata_female<-subset(mydata,sei=="f")
```

subset (データフレームのオブジェクト, 列名 == “名前”)

※ == は「等号 (等しい)」

オブジェクトから列の値が “名前” と一致するデータを抽出



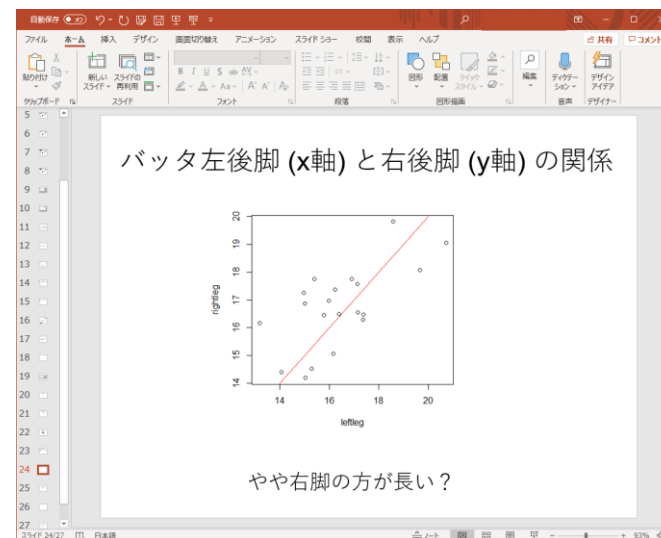
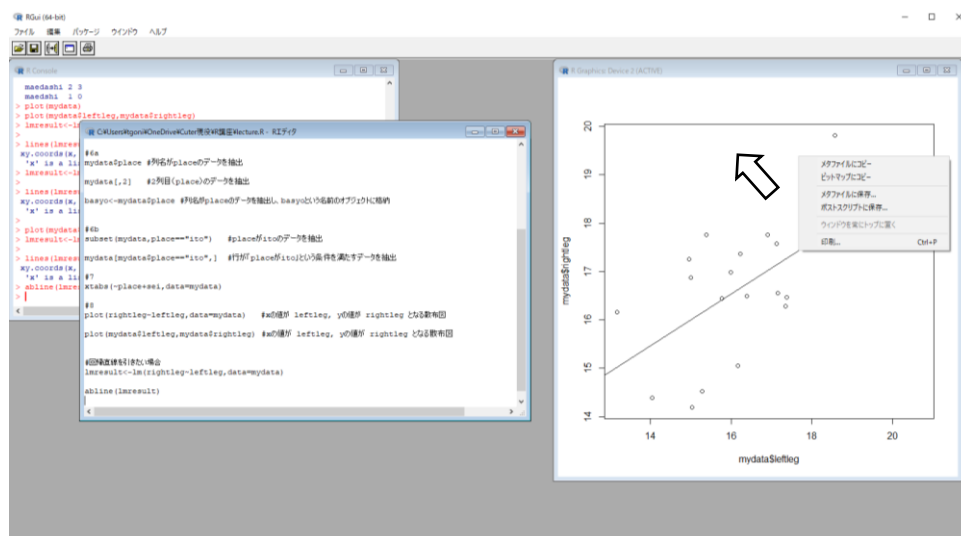
```
hist(mydata_male$bodylength)  
hist(mydata_female$bodylength)
```

画像の保存方法

図の上で右クリック > コピー



パワポやワードを開いて貼り付け



メタファイル

線画像。拡大しても見え方が変わらない。

ビットマップ

点画像。拡大すると粗くなる。

* 作図を極めたい人は
メタファイルでコピーして
[Inkscape](https://inkscape.org/ja/) というソフトに
貼り付けるのがおすすめです。

```
summary(mydata$place)
```

入力例

```
> summary(mydata$place)
  Length      Class      Mode 
    120    character character
```

要約情報が
上手く表示されない…

Rで扱う文字データには2種類の型がある

“hakozaiki”

“ito”

“maedashi”

character型

文字の集まり
(意味を持たない)

factor 型

カテゴリー
(水準の1つ
= 1つの種類を表す)

デフォルトはcharacter型→解析前にfactor型に変換しておく

as.factor() を使って character型をfactor型に変換

```
mydata$place<-as.facor(mydata$place)
mydata$sei<-as.factor(mydata$sei)
```

入力例

```
> mydata$place<-as.factor(mydata$place)
> mydata$sei<-as.factor(mydata$sei)
```

```
データフレームのオブジェクト$列名<-as.factor(データフレームのオブジェクト$列名)
```

②

①

- ① character型の列をfactor型に変換して
- ② 元の列データに代入（上書き）する

一度上書きすればそれ以降も上書きされたまま！

factor型に変換した後で再び summary() をしてみる

```
summary(mydata$place)
```

入力例

```
> summary(mydata$place)
hakozaki      ito maedashi
      43         37         40
```

今度は各水準のデータ数が表示される！

```
> summary(mydata$place)
  Length      Class      Mode 
    120 character character
```

character型の場合 (再掲載)

実験の処理や種類のデータは必ずfactor型のデータとして扱う！
(read.csv())で読み込んだ直後に変換しておくとう分かりやすい)

データは数値データとカテゴリーデータに分けられる

文字のカテゴリーデータは `as.factor()` で `factor`型に変換

1変数の解析

`summary()` でデータを要約する

1変数の数値データはヒストグラムで分布の形を見る

条件を満たす行(データ)の抽出には `subset()` を使う

2変数の関係を見る

2変数の関係を見る（視覚化しないと解釈は難しい）

ID	性	気温	体表面温度	体長	体重	採集場所
id	sei	temperature	bodytemp	bodylength	weight	place
1	f	19.66227	26.32604	29.35272	0.824753	maedashi
2	m	16.94141	19.05028	18.58453	0.401179	hakozaki
3	m	23.83568	24.29131	20.58872	0.39919	ito
4	m	24.88971	27.15717	20.23409	0.401533	ito
5	m	23.11532	24.92456	20.03364	0.399128	maedashi
6	m	15.90721	18.44454	20.18955	0.401698	maedashi
7	f	17.8181	24.37523	30.1241	0.779901	maedashi
8	m	18.70666	21.04032	17.74339	0.401306	ito
カテゴリー		数値	数値	数値	数値	カテゴリー

カテゴリー × カテゴリー → 棒グラフで視覚化

カテゴリー × 数値 → 箱ひげ図で視覚化

数値 × 数値 → 散布図で視覚化

まずは `xtabs()` で集計し、オブジェクトに格納

```
mydata_sp<-xtabs(~sei+place,data=mydata)
```

入力例

```
> mydata_sp<-xtabs(~sei+place,data=mydata)
> mydata_sp
      place
sei hakozaki ito maedashi
f      20   15     25
m      23   22     15
```

`xtabs` (集計したい列名 ~ 表の縦となる列名 + 表の横となる列名,
data =データフレームのオブジェクト)

データフレームのオブジェクトから縦と横の列の項目ごとに、
「集計したい列名」の値を合計した表データを作成

* 「集計したい列名」を書かない場合、データ数を集計

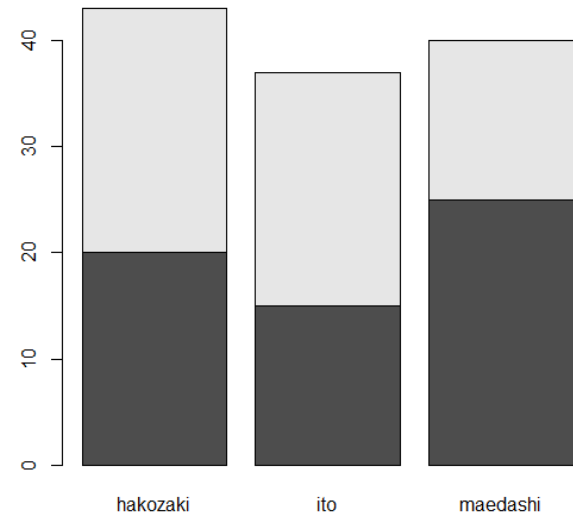
```
barplot(mydata_sp)
```

入力例

```
> barplot(mydata_sp)
```

```
barplot (表のオブジェクト)
```

表データの棒グラフを表示する



さらに引数を指定して見た目を修正する

```
barplot(mydata_sp, legend=T)
```

凡例を追加

```
barplot(mydata_sp, legend=T, args.legend=list(x=2, y=45))
```

凡例を追加して、凡例の位置を変える

```
boxplot(bodylength~sei,data=mydata)
```

入力例

```
> boxplot (bodylength~sei,data=mydata)
```

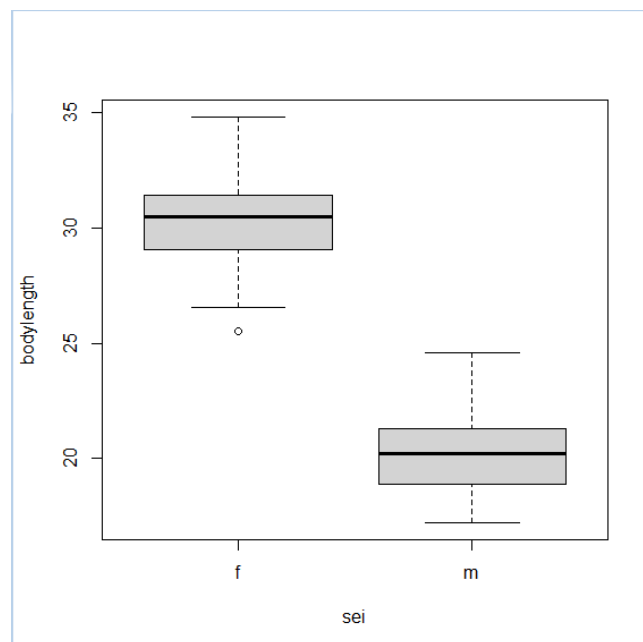
boxplot (Y軸の列名 ~ X軸の列名, data =データフレームのオブジェクト)

X軸の列(カテゴリー) と Y軸の列(数値) の箱ひげ図を作成

箱ひげ図を掲載する場合、箱とひげの説明が必要

(例：デフォルトの場合)

箱の中央線、下端、上端はそれぞれ実測値の中央値、第1四分位数、第3四分位数を表す。ひげの下端は第1四分位数から四分位範囲の1.5倍の値を引いた値以上で最小の実測値を示す。ひげの上端は第3四分位数に四分位範囲の1.5倍の値を足した値以下で最大の実測値を示す。箱とひげの範囲以外の実測値は点で表される。



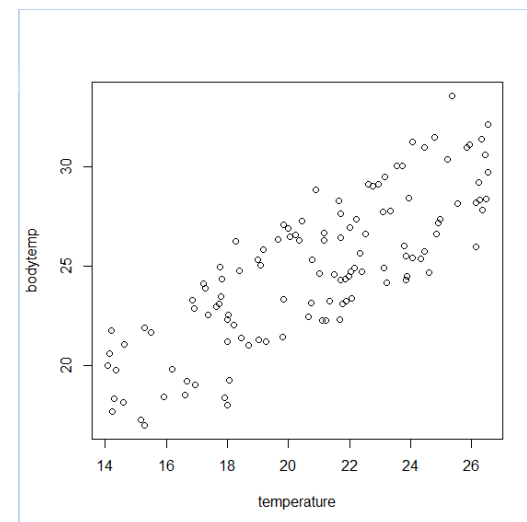
```
plot(bodytemp~temperature,data=mydata)
```

入力例

```
> plot (bodytemp~temperature,data=mydata)
```

plot (Y軸の列名 ~ X軸の列名, data =データフレームのオブジェクト)

X軸の列(数値) と
Y軸の列(数値) の散布図を作成



```
plot(bodytemp~temperature,data=mydata,col=c("red","blue")[mydata$sei])
```

性ごとに色を分ける

2変数の関係を見る | とりあえず plot() で全通り

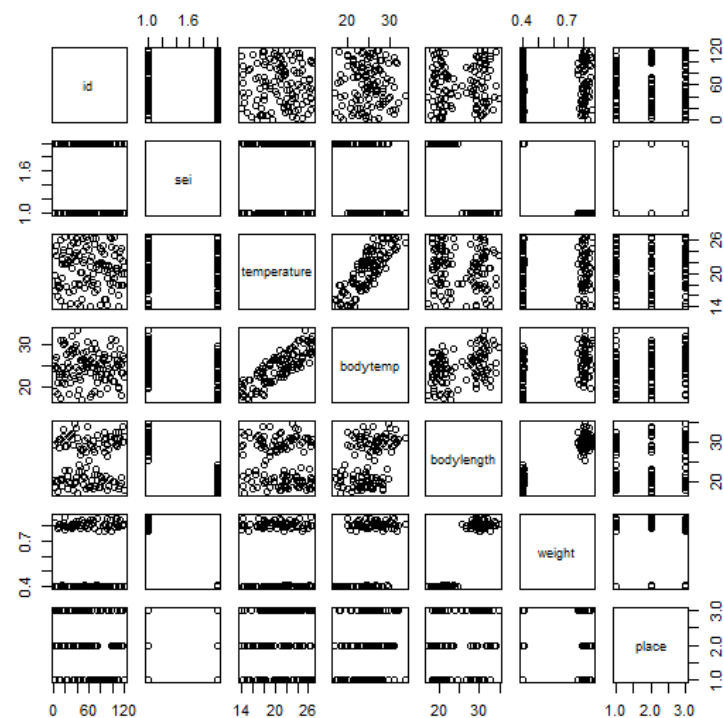
`plot(mydata)`

入力例

```
> plot(mydata)
```

plot (データフレームのオブジェクト)

列の全通りの組み合わせで
散布図を作成



2変数の解析

カテゴリー × カテゴリー

`xtabs()` を使って表に集計した後、`barplot()` で作図

カテゴリー × 数値

`boxplot()` で作図

数値 × 数値

`plot()` で作図

俯瞰するなら `plot()` で全通りを表示する

Step. 4

**Rをさらに
使いこなす**

様々な種類の統計的検定を簡単に実装できる

<code>cor()</code>	相関係数を求める
<code>t.test()</code>	t検定
<code>glm()</code>	一般化線形モデル
<code>chisq.test()</code>	カイ二乗検定
<code>fisher.test()</code>	Fisherの正確確率検定

検定結果（p値）が簡単に算出される

→何をやっているのかよく確認しておく！

線形回帰はlm()を使えばできます。

```
lmresult_male<-lm(bodytemp~temperature,data=mydata_male)
lmresult_female<-lm(bodytemp~temperature,data=mydata_female)
```

入力例
(オスの場合)

```
> lmresult_male<-lm(bodytemp~temperature,data=mydata_male)
> lmresult_male

Call:
lm(formula = bodytemp ~ temperature, data = mydata_male)

Coefficients:
(Intercept)  temperature               coefficients: (偏)回帰係数
      4.0098         0.9088             (Intercept): 切片
```

回帰直線を引くことも可能

```
plot(bodytemp~temperature,data=mydata,col=c("red","blue")[mydata$sei])
abline(lmresult_female,col="red")
abline(lmresult_male,col="blue")
```

```
mydata $ place
```

#\$ は“~の”のイメージ

の後は改行するまでコメント扱い
(実行されても計算されない)

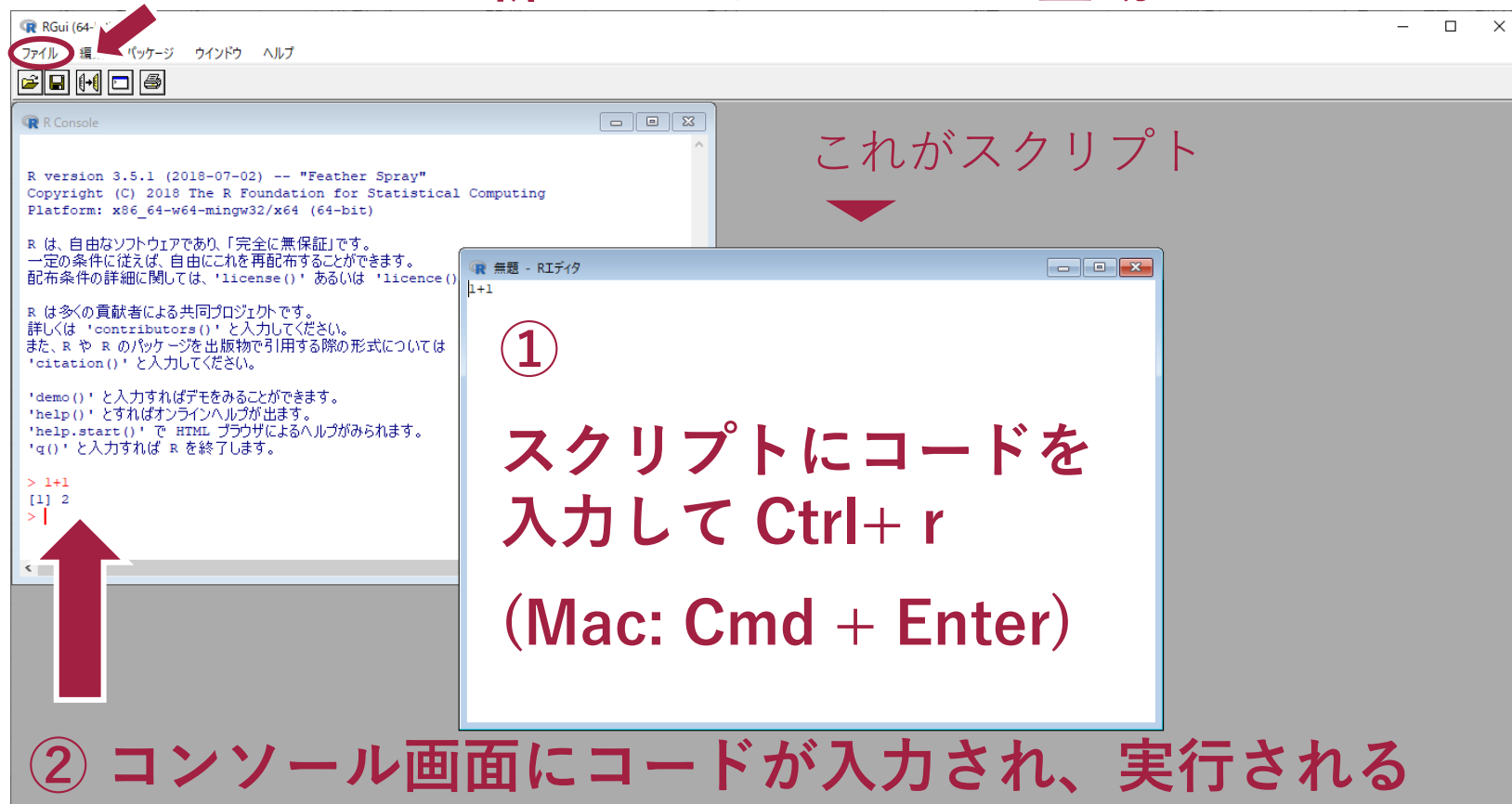


慣れるまではコメントをなるべくつける

スクリプトにコードを書く

スクリプトにコードを書いておけば、
コンソールにコピペしなくてもショートカットで実行可能！

ファイル > 新しいスクリプト で登場



The screenshot shows the RGui (64-bit) window. The 'File' menu is circled in red, with a red arrow pointing to it. The R Console window displays the R version 3.5.1 startup message and some Japanese text. A new script editor window titled '無題 - Rエディタ' is open, showing a red arrow pointing to the first line of the script. A red arrow also points to the console output area.

これがスクリプト

① スクリプトにコードを入力して Ctrl+ r
(Mac: Cmd + Enter)

② コンソール画面にコードが入力され、実行される

パッケージ = 様々な関数を定義したコード集



色々な処理を簡単にできるようになる

tidyverse	複雑なデータ整理をより簡単に
ggplot	複雑な図をより簡単に
readxl	エクセルシートを読み込める
lawstat	統計的検定のパッケージ
lme4	一般化線形混合モデル用の関数
tuneR	音データをRで扱う

install.packages(“パッケージ名”)

パッケージをPC内に保存する

該当する**パッケージをはじめて取り込むときのみ**実施

library(“パッケージ名”)

PC内のパッケージを読み込む

Rを起動するたびに実施する必要がある

パッケージについて

<https://stats.biopapyrus.jp/r/basic/package.html>

Rを使ったら引用する

citation() と

入力すれば表示される

引用文献リストの書き方

*本文中では

(R Core Team, 2021)

Mendeleyユーザーは

この部分をテキストに

貼り付け

> 拡張子 .bib で保存

> MendeleyのFile>Add File

```
R Console

'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます。
'q()' と入力すれば R を終了します。

> citation()

To cite R in publications use:

R Core Team (2021). R: A language and environment for statistical
computing. R Foundation for Statistical Computing, Vienna,
Austria. URL https://www.R-project.org/.

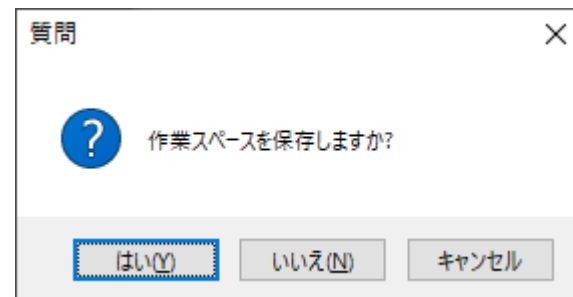
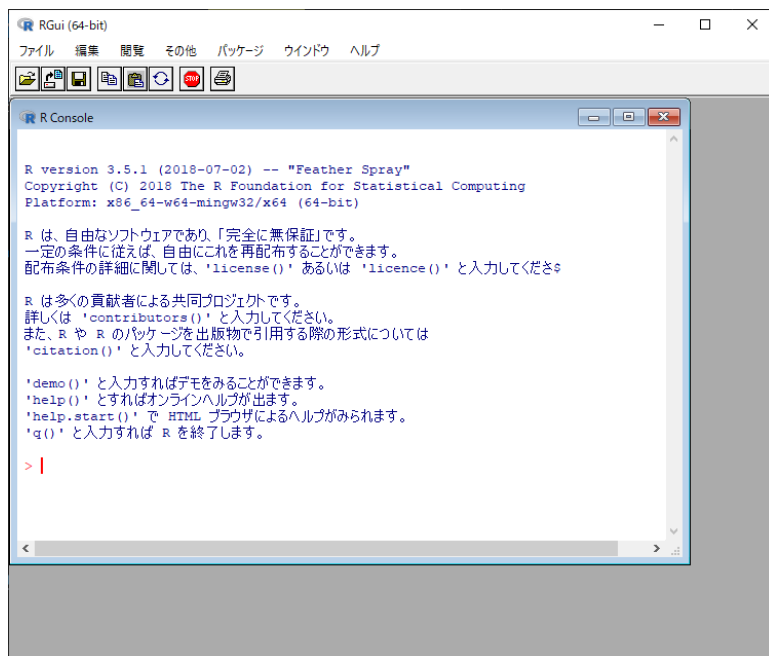
LaTeX ユーザのための BibTeX エントリーは

@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2021},
  url = {https://www.R-project.org/},
}

We have invested a lot of time and effort in creating R, please cite
it when using it for data analysis. See also 'citation("pkgname")'
for citing R packages.

> |
```

終了時
(×をクリック)



作業スペースは
保存しなくてよい

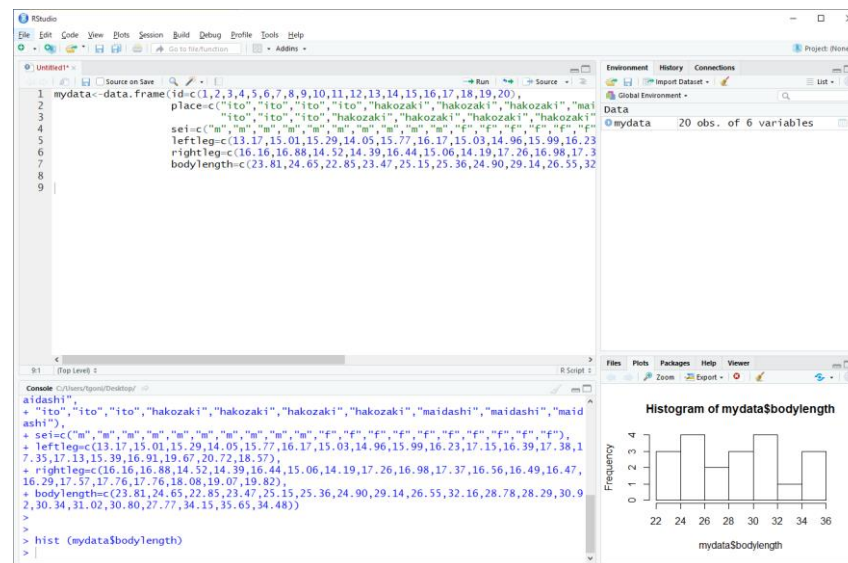
入力した内容はすべて消えるが、コードが残っていれば同じ解析はすぐにできる

RStudio : Rの操作をより簡単にするツール



<https://www.rstudio.com/products/rstudio/>

DOWNLOAD>
RStudio Desktop



無料で手に入る
(有料オプションもあるが、基本不要)

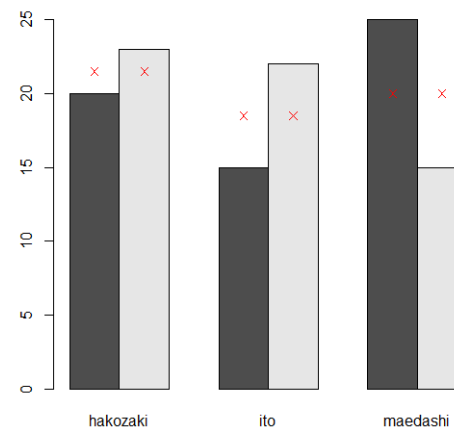
R Markdown を使えば、解析ノートが楽に作れます。

Step. Extra

事前質問

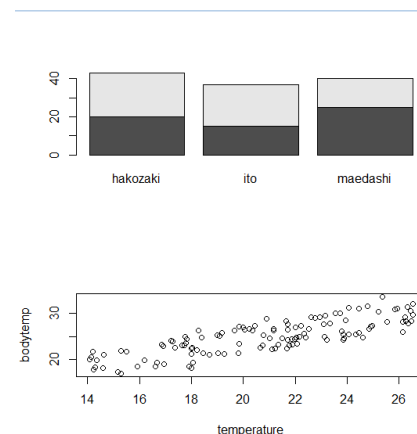
グラフに追加する→低水準関数を使う

```
barplot(mydata_sp,beside=TRUE)  
xtabs(~place,data=mydata)/2  
points(c(1.5,2.5,4.5,5.5,7.5,8.5),  
       c(21.5,21.5,18.5,18.5,20.0,20.0),  
       col="red",pch=4)
```



グラフを追加する→ par() を使って作図領域を分ける

```
par(mfrow=c(2,1))  
barplot(mydata_sp)  
plot(bodytemp~temperature,  
      data=mydata)
```



習うより慣れよ

とにかく使ってみることが一番の近道。

つまづいたらググる

フリーソフトなのでユーザーは多数存在。

検索すれば大抵の問題は解決する。

楽するために苦勞する！

最初は苦勞しますが、後で圧倒的に楽になります。

Rの悩みはRユーザーに聞くと 解決することもしばしば

Cuter に相談を！

Rユーザーいます。



中央 4F きゅうとコモンズ
理系 1F 入口付近