# データの取り扱い

データフレームとファイル

(Press `?` for help, `n` and `p` for next and previous slide)

村田 昇

2019.10.04

# データフレーム

2.1

# データ構造

- Rに用意されている基本的なデータ構造
  - ベクトル (vector): 1次元配列
  - 行列 (matrix): 2次元配列
  - 配列 (array): 多次元配列
  - **データフレーム** (data frame): 表 (2次元配列)
- 特殊なもの
  - リスト (list): オブジェクトの集合

# データフレーム

- 複数の個体について，いくつかの属性を集計した表
  - 長さの等しい列ベクトルをまとめたもの
  - 各列のデータ型はバラバラでも良い
- 例: ある小学校の1年生の身長・体重・性別・血液型のデータ
- 実データは表形式であることが多いため最も一般的な形式

2.3

# データフレームの作成

- 同じ長さのベクトルを並べる
- データフレームを結合する
- マトリクスを変換する (全て数字の場合)

```
(x <- data.frame(one=c(1,2,3),two=c("AB","CD","EF")))
x[1,2] # 1行2列の要素を選択
x[c(TRUE,FALSE,TRUE),] # 1,3行を選択
x$two # 列"two"を選択
x["two"] # 列名"two"を選択
x[-c(1,3),] # 1,3行を除外

(y <- data.frame(three=c("x","y","z"),four=c(0.9,0.5,-0.3)))
(z <- cbind(x,y))
```

2.4

# 演習

- 02-frame.r を確認してみよう

2.5

| | math | phys | chem | bio |
|---|---|---|---|---|

# 演習

- 次の表に対応するデータフレームを作成しなさい

| | math | phys | chem | bio |
|---|---|---|---|---|
| A | 90 | 25 | 65 | 70 |
| B | 80 | 50 | 100 | 50 |
| C | 70 | 75 | 70 | 30 |
| D | 60 | 100 | 40 | 80 |
| E | 50 | 80 | 75 | 100 |

# ファイルの操作

3.1

# ファイルを用いたデータの読み書き

- 解析においてはデータファイルの操作が必要:
  - 整理したデータを保存する
  - 収集されたデータを読み込む
- Rで利用可能なデータファイル:
  - CSV形式(comma separated values): テキストファイル
  - RData形式: Rの内部表現を用いたバイナリーファイル
  - (パッケージを用いればEXCELなどを扱うことも可能)

3.2

# 作業ディレクトリの確認と変更

- 作業ディレクトリとファイルに関する注意:
  - Rの処理は特定のフォルダ(**作業ディレクトリ**)内で実行される
  - ファイルは作業ディレクトリにあるものとして扱われる
  - 作業ディレクトリ以外のファイルを扱う場合はパスを含めて指定する必要がある
- 作業ディレクトリに関する操作:
  - 確認の仕方
    - コンソールの上部の表示
    - 関数 `getwd()`
  - 変更の仕方
    - “Session”>“Set Working Directory”>“Choose Directory…”
    - 関数 `setwd()`

3.3

# CSV形式の操作 (テキスト)

- 関数 `write.csv()`: CSVファイルの書き出し

```
write.csv(x, file="mydata.csv")
## x: 書き出すデータフレーム
## file: 書き出すファイルの名前 (作業ディレクトリ下, またはパスを指定)
```

- 関数 `read.csv()`: CSVファイルの読み込み

```
x <- read.csv(file="mydata.csv", header=TRUE, row.names=1)
## x: 読み込む変数
## file: 書き出すファイルの名前 (作業ディレクトリ下, またはパスを指定)
## header: 1行目を列名として使うか否か
## row.names: 行名の指定 (行名を含む列番号/列名または行名のベクトル)
```

- 他の細かいオプションはヘルプを参照

# RData形式の操作 (バイナリ)

- 関数 `save()`: RDataファイルの書き出し

```
save(..., file="mydata")
## ...: 保存するオブジェクト名（複数指定可，データフレーム以外も可）
## file: 書き出すファイルの名前（作業ディレクトリ下，またはパスを指定）
```

- 関数 `load()`: RDataファイルの読み込む

```
load(file="mydata")
## file: 読み込むファイルの名前（作業ディレクトリ下，またはパスを指定）
```

- 複数のデータフレームを同時に扱うことができる

3.5

# 演習

- 02-file.r を確認してみよう

3.6

# 演習

- 前の演習で作成したデータフレームを適当なファイルに書き出しなさい
- 書き出したファイルから別の変数に読み込みなさい

# データフレームの操作

4.1

# 部分集合の取得

- 要素を選択
  - 添字の番号を指定する (マイナスは除外)
  - 論理値(TRUE/FALSE)で指定する
  - 要素の名前で指定する

```
(x <- data.frame(one=c(1,2,3),two=c("AB","CD","EF")))
x[1,2] # 1行2列の要素を選択
x[-c(1,3),] # 1,3行を除外
x[c(TRUE,FALSE,TRUE),] # 1,3行を選択
x[,"two"] # 列名"two"を選択
```

- 関数 `subset()`: 条件を指定して行と列を選択

```
subset(x,subset,select,drop=FALSE)
## x: データフレーム
## subset: 行に関する条件
## select: 列に関する条件(未指定の場合は全ての列)
## drop: 結果が1行または1列となる場合にベクトルとして返すか否か
```

4.2

# 演習

- 02-choose.r を確認してみよう

4.3

# 演習

- `datasets::mtcars` から以下の条件を満たすデータを取り出しなさい
  - オートマチック車のデータ
  - 4気筒(`cyl`)車の燃費(`mgp`)と排気量(`disp`)のデータ
  - 馬力(`hp`)が110(馬力)以上で重さ(`wt`)が3(1000lbs)以下のデータ

# 統計量の計算

- 関数 `sum()`: 総和を計算する
- 関数 `mean()`: 平均
- 関数 `max()`: 最大値
- 関数 `min()`: 最小値
- これ以外にも沢山あるので調べてみよ

4.5

# 行・列ごとの操作

- 関数 `apply()`: 列または行ごとに計算を行う

```
apply(X, MARGIN, FUN)
## X: データフレーム
## MARGIN: 行(1)か列(2)かを指定
## FUN: 求めたい統計量を計算するための関数
```

- 関数 `aggregate()`: 各行をいくつかのグループにまとめて計算を行う

```
aggregate(x, by, FUN)
## x: データフレーム
## by: 各行が属するグループを指定するベクトルのリスト
## FUN: 求めたい統計量を計算するための関数
```

# 演習

- 02-operate.r を確認してみよう

4.7

# 演習

- `datasets::mtcars` のデータを以下の条件で整理しなさい
  - 気筒数(`cyl`)ごとに排気量(`disp`)の平均値，最大値，最小値
  - ギア数(`gear`)ごとの燃費(`mpg`)の平均値，最大値，最小値
  - 気筒数(`cyl`)とギア数(`gear`)ごとの燃費(`mpg`)の平均値

4.8